Docket No.: 62758-072 **PATENT**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Application of	:	Customer Number: 20277
Ryoichi UEDA, et al.	:	Confirmation Number:
Serial No.:	:	Group Art Unit:
Filed: February 25, 2004	:	Examiner: Unknown

For: DOCUMENT STRUCTURE INSPECTION METHOD AND APPARATUS

CLAIM OF PRIORITY AND
TRANSMITTAL OF CERTIFIED PRIORITY DOCUMENT

Mail Stop CPD
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 35 U.S.C. 119, Applicants hereby claim the priority of:

Japanese Patent Application No. 2003-140568, filed May 19, 2003

cited in the Declaration of the present application. A certified copy is submitted herewith.

Respectfully submitted,

MCDERMOTT, WILL & EMERY

Keith E. George
Registration No. 34,111

600 13th Street, N.W.
Washington, DC 20005-3096
(202) 756-8000 KEG:tlb
Facsimile: (202) 756-8087
Date: February 25, 2004

NT1460 ¥1,100
02758-072
UEDA, et al.
February 25, 2004

日本国特許庁
JAPAN PATENT OFFICE

McDermott, Will & Emery

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日
Date of Application: ２００３年　５月１９日

出願番号
Application Number: 特願２００３－１４０５６８

[ST. 10/C]: ［ＪＰ２００３－１４０５６８］

出願人
Applicant(s): 株式会社日立製作所

２００４年　１月２８日

特許庁長官
Commissioner,
Japan Patent Office
今井康夫

出証番号　出証特２００４－３００３６８７

【書類名】　特許願

【整理番号】　NT03P0254

【提出日】　平成15年 5月19日

【あて先】　特許庁長官　殿

【国際特許分類】　G06F 17/60

【発明者】

【住所又は居所】　神奈川県川崎市麻生区王禅寺１０９９番地　株式会社日立製作所　システム開発研究所内

【氏名】　植田　良一

【発明者】

【住所又は居所】　神奈川県川崎市麻生区王禅寺１０９９番地　株式会社日立製作所　システム開発研究所内

【氏名】　中山　弘二郎

【特許出願人】

【識別番号】　000005108

【氏名又は名称】　株式会社日立製作所

【代理人】

【識別番号】　100068504

【弁理士】

【氏名又は名称】　小川　勝男

【電話番号】　03-3661-0071

【選任した代理人】

【識別番号】　100086656

【弁理士】

【氏名又は名称】　田中　恭助

【電話番号】　03-3661-0071

【選任した代理人】

【識別番号】　100094352

【弁理士】

【氏名又は名称】　佐々木　孝

【電話番号】　03-3661-0071

【手数料の表示】

【予納台帳番号】　081423

【納付金額】　21,000円

【提出物件の目録】

【物件名】　明細書　1

【物件名】　図面　1

【物件名】　要約書　1

【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　文書構造検査方法および装置

【特許請求の範囲】

【請求項１】

情報処理システムにおいて送受信される文書構造定義言語で記述された構造化文書において、事前に取り決めた文書構造変化規則に基づき文書構造定義を変換し、変換後の前記文書構造定義に基づき前記構造化文書の構造を検査することにより、前記文書構造が変化した前記構造化文書の、前記文書構造変化の検査者による可逆性に依存しない検査を実現することを特徴とする文書構造検査方法。

【請求項２】

情報処理システムにおいて送受信される文書構造定義言語で記述された構造化文書において、文書構造を検査中にエラーが発生した場合、文書構造変化規則集から適用可能な文書構造変化規則を検索し、検査に利用する文書構造定義を、検索した文書構造定義に切り替えて検査を続行することにより、前記文書構造が変化した前記構造化文書の、前記文書構造変化の検査者による可逆性に依存しない検査を実現することを特徴とする文書構造検査方法。

【請求項３】

請求項１または２に記載の文書構造検査方法において、文書構造定義言語毎に文書構造検査部を準備し、文書構造定義の記述に利用している文書構造定義言語に応じて、文書構造検査部を入れ替えることにより、前記文書構造定義が複数の言語により記述されている場合でも検査可能とすることを特徴とする文書構造検査方法。

【請求項４】

情報処理システムにおいて送受信される文書構造定義言語で記述された構造化文書において、文書構造変化規則集に記憶された文書構造変化規則に従って、文書構造定義集に記憶された文書構造定義を変換する手段と、該変換手段により変換された文書構造定義を利用して文書構造を検査する手段とを有することを特徴とする文書構造検査装置。

【請求項５】

情報処理システムにおいて送受信される文書構造定義言語で記述された構造化文書において、文書構造を検査する手段と、前記文書構造の検査中にエラーが発生した場合文書構造変化規則集から適用可能な文書構造変化規則を検索する手段と、検査中に検査に利用する文書構造定義を切り替える手段とを有することを特徴とする文書構造検査装置。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、構造化文書の構造検査方法に関する。

【０００２】

【従来の技術】

一般に通信ネットワークを介して処理要求を受け取り、要求に見合った処理を行う計算機システムでは、要求内容を表す文書（ネットワーク上で送受信される電子データを意味する）を受け取るたびにそのデータ構造が事前に取り決めた構造と合致するかどうかを検査し、文書の構造が正しい場合にのみ主となる業務処理を行い、その結果を返信する。文書の構造が正しくない場合はエラーを返信する。

【０００３】

このような文書構造検査処理は業務処理を実行しながら徐々に実施することも可能であるが、業務処理の最終段階になって文書構造が正しくないことがわかった場合、それ以前の全ての業務処理が無駄になってしまい、処理効率が悪化する。このような処理効率の悪化を防ぐために、業務処理の前に文書構造全体の検査が行われることが望ましい。これは送受信される情報がＸＭＬ（Ｅｘｔｅｎｓｉｂｌｅ　Ｍａｒｋｕｐ　Ｌａｎｇｕａｇｅ）で表現されている場合も同様である。

【０００４】

個々の構造化文書の構造があらかじめ定義された文書構造定義と合致していることを検査する処理は、Ｊａｖａ(登録商標)を対象として、Ｒａｊｉｖ　Ｍｏｒｄａｎｉ他が、「Ｊａｖａ　ＡＰＩ　ｆｏｒ　ＸＭＬ　Ｐｒｏｃｅｓｓｉｎｇ

Ｖｅｒｓｉｏｎ　１．２　Ｆｉｎａｌ　Ｒｅｌｅａｓｅ」（非特許文献１）などで「Ｓｃｈｅｍａ　Ｖａｌｉｄａｔｉｏｎ」としてそのインターフェイスを規定しており、Ｔｈｅ　Ａｐａｃｈｅ　Ｓｏｆｔｗａｒｅ　Ｆｏｕｎｄａｔｉｏｎ、「Ｘｅｒｃｅｓ２　Ｊａｖａ　Ｐａｒｓｅｒ　Ｒｅａｄｍｅ」（非特許文献２）などでは、既に実装されていることが述べられている。

【０００５】

また、前記のような通信ネットワークを介して情報を受け取る計算機システムでは、送信文書の改竄防止や盗聴防止などの目的で送信データに送信者の電子署名を付与したり、送信文書を暗号化するといった事が行われる場合がある。

【０００６】

さらに、複数の計算機が順番にそれぞれの処理を行い、全体としてより複雑な処理を実現する計算機システムの場合には、最初の送信者（処理要求者）は送信データの一部分に電子署名を付与したり、一部分を暗号化したりすることがある。

【０００７】

このような処理を行う例として、オンラインショッピングがある。オンラインショッピングでクレジットカードを利用して支払いを行う場合、購入者（最初の送信者）からオンラインショッピングサイト運営者に送信されるデータには、購入するものを特定するための情報と、支払いに利用するクレジットカードを特定するための情報（カード番号、有効期限など）が含まれる。このうちクレジットカードに関する情報がオンラインショッピング運営者からクレジットカード会社に送信され、実際のクレジット決済処理が行われる。オンラインショッピング運営者はクレジットカード情報を直接利用するわけではないが、購入者から受け取ってクレジットカード会社に中継する役割を担う。

【０００８】

このようなシステムでは、送信データ全体をオンラインショッピング運営者が復号化できる方式で送信する方式よりも、利用者側でクレジットカード会社だけがクレジットカード情報を復号化できる方式で暗号化（部分暗号）し、オンラインショッピングサイト運営者にはクレジットカード情報を開示しない方式の方が

安全性が高い。また、この時、クレジットカード情報が購入者自身により暗号化されたことを示すために暗号化後のデータに電子署名（部分署名）を付与することがある。

【０００９】

今後は公開鍵暗号基盤の普及によりこのような電子取引が主流になると考えられている。

【００１０】

一般にある文書構造定義言語で記述された文書構造定義を別の言語で書かれた等価な文書構造定義に変換する技術が存在する。例えば、非特許文献３には文書定義言語ＤＴＤで記述された文書構造定義を「Ｗ３Ｃ　ＸＭＬ　Ｓｃｈｅｍａ」で記述された等価な文書構造定義に変換するツール「ｄｔｄ２ｘｓ」が紹介されている。ただし、文書構造定義言語の記述能力の違いなどに起因して、あらゆる文書構造定義言語間の変換で完全に等価な文書構造定義が得られると言うわけではない。

【００１１】

【非特許文献１】

Ｒａｊｉｖ　Ｍｏｒｄａｎｉ、他１名、「Ｊａｖａ　ＡＰＩ　ｆｏｒ　ＸＭＬ　Ｐｒｏｃｅｓｓｉｎｇ　Ｖｅｒｓｉｏｎ　１．２　Ｆｉｎａｌ　Ｒｅｌｅａｓｅ」、［ｏｎｌｉｎｅ］、２００３年２月１０日、［同日検索］、インターネット＜ＵＲＬ：ｈｔｔｐ：／／ｗｗｗ．ｊｃｐ．ｏｒｇ／ａｂｏｕｔＪａｖａ／ｃｏｍｍｕｎｉｔｙｐｒｏｃｅｓｓ／ｆｉｎａｌ／ｊｓｒ０６３／ｉｎｄｅｘ２．ｈｔｍｌ＞

【非特許文献２】

Ｔｈｅ　Ａｐａｃｈｅ　Ｓｏｆｔｗａｒｅ　Ｆｏｕｎｄａｔｉｏｎ、「Ｘｅｒｃｅｓ２　Ｊａｖａ　Ｐａｒｓｅｒ　Ｒｅａｄｍｅ」、［ｏｎｌｉｎｅ］、２００２年、［２００３年２月１０日検索］、インターネット＜ＵＲＬ：ｈｔｔｐ：／／ｘｍｌ．ａｐａｃｈｅ．ｏｒｇ／ｘｅｒｃｅｓ２－ｊ／ｉｎｄｅｘ．ｈｔｍｌ＞

【非特許文献３】

ＬｕＭｒｉＸ、「ＸＭＬ　Ｔｏｏｌｓ」、［ｏｎｌｉｎｅ］、２００３年、［２００３年2月１４日検索］、インターネット＜ＵＲＬ：ｈｔｔｐ：／／ｐｕｖｏｇｅｌ．ｉｎｆｏｒｍａｔｉｋ．ｍｅｄ．ｕｎｉ－ｇｉｅｓｓｅｎ．ｄｅ／ｌｕｍｒｉｘ／＃ｄｔｄ＞

【００１２】

【発明が解決しようとする課題】

従来技術では、前記のように送受信される構造化文書に署名付与、暗号化などの処理を施すと文書の構造が変化するため、署名付与、暗号化などの処理を施さない文書と同じ方法で構造検査を実施することができない、という問題点がある。署名付与だけの場合、署名全体を除去してから構造検査を行うようにすることも可能だが、上記暗号化されたクレジットカード情報を受け取ったオンラインショッピング運営者のように、自身では復号化できない文書を受け取った場合、構造の検査が実施できなくなる。

【００１３】

本発明の目的は、構造化文書の全部または一部に署名が付与される、または、構造化文書の全部または一部が暗号化されるなどの理由により文書構造が変化する場合でも、また、暗号化部分を検査者が復号化できない場合でも、文書全体の構造検査を実施可能とすることにある。

【００１４】

【課題を解決するための手段】

本発明では、文書に対する署名付与の場所と方式と署名データの構造、暗号化対象のデータと暗号化方式など文書構造を変化させる処理の内容をすべて事前に取り決め、文書構造検査時に利用される文書構造定義を検査前に署名付与／暗号化後の形式に変換し、該変換後の文書構造定義を利用して従来どおりの文書構造検査を実施することにより前記目的を達成する。

また、本発明では別の方法として、文書構造検査部が署名付与および暗号化による文書構造の変化規則を読み込み、さらに文書構造検査時に該変化規則を動的に適用するように文書構造検査部を拡張して、送信者の署名の有無／方式、暗号化箇所／方式などを認識することにより、前記目的を達成する。

【００１５】

【発明の実施の形態】

以下の説明では構造化文書の、署名および暗号化による構造の変化に対応した文書構造検査方法および装置について述べているが、本発明は署名および暗号化以外の操作による文書構造変化にも対応可能である。

【００１６】

以下、本発明の第１の実施例を図により説明する。

本発明は図１に示したような複数の計算機１０１、１０２、１０４がネットワーク１０３（有線または無線）により接続され、相互に情報のやり取りを行う環境において、情報の受信者が、受信データ構造が事前に取り決めたものと一致しているかどうかを検査する方法およびシステムに関するものである。

【００１７】

本実施例では２者間の情報のやり取りを例にとって説明するが、本発明は一連の処理に3者以上が関わる通信に含まれる任意の２者間通信にも適用可能である。すなわち、ＡからＢを経てＣに情報が送られるような３者が関わる通信の場合、ＡからＢへの送信、および、ＢからＣへの送信の２箇所で適用可能である。

【００１８】

図２は受信者側の計算機の構成を示したものである。受信者側の計算機１０４はＣＰＵ２０１、メモリ２０２、表示装置２０３、入力装置２０４、通信装置２０５、記憶装置２０６からなり、ネットワーク１０３に接続されている。該記憶装置２０６には受信文書の構造を検査する文書構造検査部２１１、受信文書に含まれる業務要求を処理する業務処理部２１２、受信文書の構造を記述した文書構造定義を記憶する文書構造定義集２１３、署名や暗号化など、送信文書に対する操作によりどのように文書構造が変化するかを記述した文書構造変化規則を記憶する文書構造変化規則集２１５、前記文書構造変化規則に従って前期文書構造定義を変化させる文書構造定義変換部２１４が含まれる。

【００１９】

本発明は記憶装置に格納された前記処理プログラム（文書構造検査部、業務処理部、文書構造定義変換部）がメモリに読み込まれ、ＣＰＵで実行されることに

より実現される。

【００２０】

本実施例の前記文書構造検査部２１１は非特許文献２で既に実現されているものと同じものである。すなわち、受信した構造化文書が対応する前記文書構造定義と合致するかどうかを検査する。この検査で合致した場合、前記業務処理部２１２で該受信文書の要求に従って業務処理を実行する。文書構造検査に先立って前記文書構造変換部２１４が前記文書構造変化規則に従って前記文書構造定義の変換処理を行う。前記文書構造検査部は該変換後の文書構造定義を利用して、検査を実施する。

【００２１】

図３は前記文書構造定義集２１３に記録される文書構造定義の一例を、ＤＴＤ（Ｄｏｃｕｍｅｎｔ　Ｔｙｐｅ　Ｄｅｆｉｎｉｔｉｏｎ）を利用して示した例である。ＤＴＤはＸＭＬ文書などの文書構造を定義するために設計された文書構造定義言語である。ここではＤＴＤを利用した場合の例を示したが、他の文書構造定義言語による定義でも構わない。ＸＭＬ文書は入れ子関係を持つ要素の集合で、その先頭の要素を「ルート要素」と呼ぶ。要素Ｅが要素Ｆに含まれる場合、要素Ｅと要素Ｆの関係を「親子関係」と呼び、要素Ｅは要素Ｆの「親要素」、要素Ｆは要素Ｅの「子要素」と呼ぶ。同じ親要素を持つ要素同士を「兄弟要素」と呼ぶ。各要素はそれぞれ０個以上の属性を持つことができる。

【００２２】

図３には説明のために文書構造定義の各行の先頭に２桁の行番号を付与しているが、実際の文書構造定義には行番号は含まれない。同様に、以降の説明で利用する構造化文書、文書構造定義には説明のために行番号を付与しているが実際の構造化文書、文書構造定義にはこの番号は含まれない。

【００２３】

図３の１行目（行番号０１）はルート要素が「ＰｕｒｃｈａｓｅＯｒｄｅｒ」であることを表す。２行目（行番号０２）は前記「ＰｕｒｃｈａｓｅＯｒｄｅｒ」要素が「ＵｓｅｒＩＤ」、「Ｐｒｉｃｅ」、「ＣｒｅｄｉｔＣａｒｄ」の３つの子要素を持ち、これらがこの順で１回だけ出現することを表す。３行目（行番

号０３）は前記「ＰｕｒｃｈａｓｅＯｒｄｅｒ」要素が「Ｉｄ」という属性を持ち、該属性が存在する場合、その値がＸＭＬ文書中で一意でないといけないこと、該属性は存在しなくても良いことを表す。４行目（行番号０４）、５行目（行番号０５）はそれぞれ「ＵｓｅｒＩＤ」要素、「Ｐｒｉｃｅ」要素の値として任意の文字列が許容されることを表す。６行目（行番号０６）は「ＣｒｅｄｉｔＣａｒｄ」要素が「Ｉｓｓｕｅｒ」、「Ｎｕｍｂｅｒ」、「Ｅｘｐｉｒｅ」、「Ｏｗｎｅｒ」の４つの子要素を持ち、これらがこの順で１回だけ出現することを表す。７行目（行番号０７）、８行目（行番号０８）、９行目（行番号０９）、１０行目（行番号１０）はそれぞれ「Ｉｓｓｕｅｒ」要素、「Ｎｕｍｂｅｒ」要素、「Ｅｘｐｉｒｅ」要素、「Ｏｗｎｅｒ」要素の値として、任意の文字列が許容されることを表す。ここでは、本文書構造定義を「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」と名付けて前記文書構造定義集２１３に記憶しているものとする。

【００２４】

図４は前記文書構造定義「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」に合致するＸＭＬ文書の一例である。ただし、ここでは説明をわかりやすくするために、通常のＸＭＬ文書に含まれる名前空間を省略している。名前空間はひとつのＸＭＬ文書内で要素名を一意に保つために利用されるもので、本例では名前空間なしでも要素名が一意になるようにしているため名前空間を省略しても問題ない。

【００２５】

図４の１行目（行番号０１）は本文書がＸＭＬ文書であることを表し、２行目（行番号０２）は本ＸＭＬ文書の構造が「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」で定義されたものに合致することを表す。３行目（行番号０３）以降が実際の文書を表す。４行目（行番号０４）はＵｓｅｒＩＤ要素の値が「１０１９４９７０」であることを、５行目（行番号０５）はＰｒｉｃｅ要素の値が「１００００」であることを表す。

【００２６】

図５は前記文書構造変化規則集２１５に記録される文書構造変化規則の例である。ここでは３つの規則５１１、５１２、５１３が挙げられており、それぞれの規則は、変換の方法を示す種別５０１、変換対象の文書構造定義と適用箇所を示

す適用場所５０２、変化後の操作要素５０３、変化後の文書構造定義の元となる関連文書構造定義５０４および文書構造変化規則番号５０５からなる。

【００２７】

前記種別５０１には「Ｒｅｐｌａｃｅ」「Ａｄｄ」「Ｄｅｌｅｔｅ」の３種類がある。「Ｒｅｐｌａｃｅ」は前記適用場所５０２で示した要素を前記操作要素５０３で示した要素で置換することを、「Ａｄｄ」は前記適用場所５０２で指定した場所に前記操作要素５０３を追加することを、「Ｄｅｌｅｔｅ」は前記適用場所５０２で示した要素が削除されることを表す。種別が「Ｄｅｌｅｔｅ」である場合、操作要素５０３および関連文書構造定義５０４は「空」としても構わない。

【００２８】

前記適用場所５０２は、文書構造定義とその定義内の場所を示す文字列を「：」記号で連結した文字列で表現される。該文書構造定義内の場所は木構造のルートから順に指定要素までの要素名を、「／」記号を区切り文字として連結した文字列で表現される。

【００２９】

また、ある要素を指定するのではなく木構造の特定の位置を指定するために、「ある要素の先頭の子要素のさらに前」を意味する「ｆｉｒｓｔ（）」や、「ある要素の末尾の子要素のさらに後」を意味する「ｌａｓｔ（）」、「ある要素の直前（ある要素Ｅの直前に兄弟要素Ｄが存在する場合ＥとＤの間）」を意味する「ｂｅｆｏｒｅ（）」、「ある要素の直後（ある要素Ｅの直後に兄弟要素Ｆが存在する場合ＥとＦの間）」を意味する「ａｆｔｅｒ（）」を利用して表現しても良い。「ある要素の値」を意味する「ｖａｌｕｅ（）」を利用して表現しても良い。さらに、連続する兄弟要素を記号「－」で連結して、範囲を表現することも可能である。例えば「ｐｏ．ｄｔｄ：／ｐｏ／ｖａｌｕｅ（）」は文書構造定義「ｐｏ．ｄｔｄ」のルート要素「ｐｏ」の値を、「ｐｏ．ｄｔｄ：／ｐｏ／ｏｎｅ－ｔｈｒｅｅ」は文書構造定義「ｐｏ．ｄｔｄ」のルート要素「ｐｏ」の連続する子要素「ｏｎｅ」から「ｔｈｒｅｅ」までの範囲を表す。

【００３０】

例えば、1番目の文書構造変化規則５１１の適用場所５０２「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ：／ＰｕｒｃｈａｓｅＯｒｄｅｒ／ＣｒｅｄｉｔＣａｒｄ」は文書構造定義「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」のルート要素「ＰｕｒｃｈａｓｅＯｒｄｅｒ」の子要素「ＣｒｅｄｉｔＣａｒｄ」を表す。また、２番目の文書構造変化規則５１２の適用場所５０２「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ：／ＰｕｒｃｈａｓｅＯｒｄｅｒ／ｌａｓｔ（）」は文書構造定義「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」のルート要素「ＰｕｒｃｈａｓｅＯｒｄｅｒ」の最後の子要素のさらに後を表す。ここでは特殊な表記法を用いたが、文書構造定義の特定の場所を指し示すことができる表記法であれば上記以外の方法で表現しても良い。

【００３１】

前記操作要素５０３も前記適用場所と同じ表記方法で置換または追加する要素を表現する。例えば、1番目の文書構造変化規則５１１の操作要素「ＥｎｃｒｙｐｔｅｄＤａｔａ．ｄｔｄ：／ＥｎｃｒｙｐｔｅｄＤａｔａ」は文書構造定義「ＥｎｃｒｙｐｔｅｄＤａｔａ．ｄｔｄ」のルート要素「ＥｎｃｒｙｐｔｅｄＤａｔａ」を表す。

【００３２】

前記関連文書構造定義５０４は文書構造の変化時に必要となる文書構造定義を表す。複数の文書構造定義をあげることができる。例えば、１番目の文書構造変化規則５１１の関連文書構造定義「ＥｎｃｒｙｐｔｅｄＤａｔａ．ｄｔｄ　ＫｅｙＩｎｆｏ．ｄｔｄ」は「ＥｎｃｒｙｐｔｅｄＤａｔａ．ｄｔｄ」と「ＫｅｙＩｎｆｏ．ｄｔｄ」の２つの文書構造定義が該文書構造変化規則に関連していることを示す。該関連文書構造定義５０４であげた文書構造定義を含めて、前記文書構造変化規則集２１５内に出現する全ての文書構造定義を事前に入手し、前記文書構造定義集２１３にあらかじめ記憶していなければならない。

【００３３】

図６はＥｎｃｒｙｐｔｅｄＤａｔａ．ｄｔｄの例、図７はＥｎｃｒｙｐｔｅｄＫｅｙ．ｄｔｄの例、図８はＳｉｇｎａｔｕｒｅ．ｄｔｄの例、図９はＫｅｙＩｎｆｏ．ｄｔｄの例である。それぞれはＷ３Ｃ（Ｗｏｒｌｄ　Ｗｉｄｅ　Ｗｅｂ

Ｃｏｎｓｏｒｔｉｕｍ）で策定されたＸＭＬ文書の暗号化に関する仕様（ＸＭＬ　Ｅｎｃｒｙｐｔｉｏｎ）および、ＸＭＬ文書の署名に関する仕様（ＸＭＬ　Ｓｉｇｎａｔｕｒｅ）を元に、説明のために簡単化して記述したものである。

【００３４】

図１０は前記文書構造定義変換部２１４の処理手順を表す。はじめに全ての文書構造変化規則を適用済みかどうかを検査し（ステップ１００１）、全て適用済みなら処理を終了する。未適用の変化規則がある場合、次の変化規則を取得し（ステップ１００２）、種別を調べる。種別が「Ｒｅｐｌａｃｅ」の場合（ステップ１００３）、該変化規則の適用場所に記載の要素を操作要素で置換する（ステップ１０１０）。種別が「Ａｄｄ」の場合（ステップ１００４）、該変化規則の適用場所に操作要素を追加する（ステップ１０１１）。

さらに、前記置換処理（ステップ１０１０）および追加処理（ステップ１０１１）にはこれに伴って必要となる、前記文書構造定義集２１３への、置換および追加要素とその定義に利用される要素の構造定義の追加も行う（ステップ１０１２）。

【００３５】

図５の３つの文書構造変化規則５１１、５１２、５１３を適用する例を以下に示す。

図３の文書構造定義「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」に、1番目の文書構造変化規則５１１を適用した後の状態を図１１に示す。1番目の文書構造変化規則５１１は種別が「Ｒｅｐｌａｃｅ」、適用場所が「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ：／ＰｕｃｈａｓｅＯｒｄｅｒ／ＣｒｅｄｉｔＣａｒｄ」、操作要素が「ＥｎｃｒｙｐｔｅｄＤａｔａ．ｄｔｄ：／ＥｎｃｒｙｐｔｅｄＤａｔａ」なので、「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」のルート要素「ＰｕｒｃｈａｓｅＯｒｄｅｒ」の子要素「ＣｒｅｄｉｔＣａｒｄ」（図3の３２０）を「ＥｎｃｒｙｐｔｅｄＤａｔａ．ｄｔｄ」のルート要素「ＥｎｃｒｙｐｔｅｄＤａｔａ」（１１２１）で置換する。変換前、ルート要素「ＰｕｒｃｈａｓｅＯｒｄｅｒ」は「ＵｓｅｒＩＤ」「Ｐｒｉｃｅ」「ＣｒｅｄｉｔＣａｒｄ」の３つの子要素を持つと定義されている（図３の行番号０２）。これが変換により「ＵｓｅｒＩ

D」「Ｐｒｉｃｅ」「ＥｎｃｒｙｐｔｅｄＤａｔａ」の３つの子要素を持つようになる（図１１の行番号０２）。

【００３６】

さらに、１番目の文書構造変化規則５１１の関連文書構造定義にあげられた２つの文書構造定義「ＥｎｃｒｙｐｔｅｄＤａｔａ．ｄｔｄ」および「ＫｅｙＩｎｆｏ．ｄｔｄ」を適用対象となる文書構造定義「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」に追加するのが図１１の行番号１１～１６（１１１４）、及び図１１の行番号１７～２２（１１１５）である。この例ではこれらの文書構造定義の実体を追加しているが、対応する文書構造定義の参照（文書構造定義を一意に特定するための情報）のみを追加し、実体を前記文書構造定義集２１３に「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」とは分離して記憶する方式でも良い。図２２に示した文書構造定義２２０１は、図１１に示した文書構造定義１１０１を文書構造定義「ＥｎｃｒｙｐｔｅｄＤａｔａ．ｄｔｄ」と「ＫｅｙＩｎｆｏ．ｄｔｄ」への参照を用いて表現した例である。図１１の行番号１１～１６（１１１４）に対応する部分を参照を用いて表現したのが図２２の行番号１１～１２（２２１１）、図１１の行番号１７～２２（１１１５）に対応する部分を参照を用いて表現したのが図２２の行番号１３～１４（２２１２）である。

【００３７】

また、要素の置換処理により「ＣｒｅｄｉｔＣａｒｄ」要素は不要となるためその定義（行番号０６）および、「ＣｒｅｄｉｔＣａｒｄ」の定義でのみ利用される他の要素（図１１の行番号０７～１０（１１１３））は文書構造定義「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」から削除しても良い。

【００３８】

次に、さらに2番目の文書構造変化規則５１２を適用した後の文書構造定義「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」を図１２に示す。2番目の文書構造変化規則５１２は種別が「Ａｄｄ」、適用場所が「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ：／ＰｕｃｈａｓｅＯｒｄｅｒ／ｌａｓｔ（）」、操作要素が「ＥｎｃｒｙｐｔｅｄＫｅｙ．ｄｔｄ：／ＥｎｃｒｙｐｔｅｄＫｅｙ」なので、「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」のルート要素「ＰｕｒｃｈａｓｅＯｒｄｅｒ」の最

後の要素として「ＥｎｃｒｙｐｔｅｄＫｅｙ．ｄｔｄ」のルート要素「ＥｎｃｒｙｐｔｅｄＫｅｙ」を追加する（図１２の行番号０２の１２１１）。さらに、２番目の文書構造変化規則の関連文書構造定義にあげられた文書構造定義「ＥｎｃｒｙｐｔｅｄＫｅｙ．ｄｔｄ」を適用対象となる文書構造定義「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」に追加する（図１２の行番号２３～２９（１２１２））。

【００３９】

最後に、同様にしてさらに3番目の文書構造変化規則５１３を適用した後の文書構造定義「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」を図１３に示す。「ＰｕｒｃｈａｓｅＯｒｄｅｒ」要素の最後の要素として「Ｓｉｇｎａｔｕｒｅ．ｄｔｄ」のルート要素「Ｓｉｇｎａｔｕｒｅ」を追加し（行番号０２の１３１１）、「Ｓｉｇｎａｔｕｒｅ」要素の構造定義（行番号３０～４１の１３１２）を追加している。

【００４０】

以上で文書構造変化規則集２１５の全ての文書構造変化規則を適用したため、前記文書構造変換部２１４の処理を終了する。

【００４１】

上記の文書構造変換処理により、図４に示した構造化文書を暗号化し、署名を付与した後の文書（図１４）の構造検査が可能となる。図１４の行番号０６～１５（１４１１）は「ＣｒｅｄｉｔＣａｒｄ」要素を暗号化した結果、行番号１６～２７（１４１２）は暗号に利用した鍵の情報、行番号２８～４１（１４１３）は署名を表す。

【００４２】

上記では文書構造変化規則として、「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」の構造が変化するもののみをあげたが、図１５に示すような「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」以外の文書構造定義の構造を変化させる文書構造変化規則を定義することも可能である。

【００４３】

図１５の１５１１は署名の方式の一つで、署名対象文書を署名の中に取り込む

方式（一般に「Ｅｎｖｅｌｏｐｉｎｇ署名」と呼ばれている）に対応させる文書構造変化規則の例である。

【００４４】

これを適用した場合の文書構造定義の変化を図１６に示す。文書構造変化規則１５１１は種別が「Ａｄｄ」、適用場所が「Ｓｉｇｎａｔｕｒｅ．ｄｔｄ：／Ｓｉｇｎａｔｕｒｅ／ｌａｓｔ（）」、操作要素が「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ：／ＰｕｒｃｈａｓｅＯｒｄｅｒ」なので、文書構造定義「Ｓｉｇｎａｔｕｒｅ．ｄｔｄ」のルート要素「Ｓｉｇｎａｔｕｒｅ」の最後の子要素として、文書構造定義「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」のルート要素「ＰｕｒｃｈａｓｅＯｒｄｅｒ」を行番号０２に追加（１６１１）、さらに、関連文書構造定義にあげられた「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」と「ＫｅｙＩｎｆｏ．ｄｔｄ」を「Ｓｉｇｎａｔｕｒｅ．ｄｔｄ」に行番号１４～２２（１６１２）、及び行番号２３～２８（１６１３）で追加する。該文書構造変化規則を適用すると受信文書のルート要素は「Ｓｉｇｎａｔｕｒｅ」に変更されることになる。

【００４５】

上記のように、本発明の第１の実施例によれば、構造化文書の全部または一部に署名が付与される、または、構造化文書の全部または一部が暗号化されるなどの理由により文書構造が変化する場合でも、また、暗号化部分を検査者が復号化できない場合でも、文書全体の構造検査を実施することが可能となる。

【００４６】

以下、本発明の第２の実施例を図により説明する。

図１７に本発明の第２の実施例のハードウェア構成を示す。本発明の第２の実施例のハードウェア構成は、第１の実施例と同じである。違いは記憶装置内の文書構造検査部１７０１での処理方法、および、文書構造変化規則集１７１１のデータ構造にある。

【００４７】

図１８は前記文書構造変化規則集１７１１に記録される文書構造変化規則の例である。ここでは４つの規則１８１１、１８１２、１８１３、１８１４があげら

れており、それぞれの規則は、変換の方法を示す種別１８０１、適用文書構造定義を表す適用先１８０２、変換操作の対象となる操作要素１８０３、変化後の文書構造検査に必要となる関連文書構造定義１８０４および文書構造変化規則番号１８０５からなる。

【００４８】

前記種別１８０１には「Ｒｅｐｌａｃｅ」「Ａｄｄ」の２種類がある。「Ｒｅｐｌａｃｅ」は前記適用先１８０２の文書構造定義の一部が前記操作要素１８０３で示した文書構造定義の要素に置換される可能性があることを、また「Ａｄｄ」は前記適用先１８０２の文書構造定義の一部に前記操作要素１８０３で示した文書構造定義の要素が追加される可能性があることを表す。前記適用先１８０２には文書構造定義名または記号「＊」が記述される。「＊」は適用先としてどの文書構造定義でも良いことを表す。前記操作要素１８０３および関連文書構造定義１８０４の表記法はそれぞれ第1の実施例で示した図５の操作要素５０３および関連文書構造定義５０４に準ずる。

【００４９】

図１９は前記文書構造検査部１７０１の処理手順を表す。はじめに検査対象となる構造化文書に記載された対応文書構造定義を特定し、現文書構造定義に設定する（ステップ１９０１）。検査対象構造化文書の検査位置および文書構造定義の検査位置は共に先頭に設定する。次に、現文書構造定義と検査対象構造化文書が合致するかどうかを検査位置から順番に従来通り検査する（ステップ１９０２）。ステップ１９０２は検査が文書構造定義の末尾に到達する、もしくは、検査対象構造化文書の末尾に到達する、もしくはエラーが発生する、のいずれかの場合に停止してステップ１９０３に処理が移る。ここで「文書構造定義の末尾」とは文書構造定義の最終行を意味するわけではなく、「当該文書構造定義のルート要素の定義の終端」を意味する。

【００５０】

ステップ１９０３では検査が文書構造定義の末尾まで到達したかどうかを調べ、検査が末尾に到達しなかった場合、異常終了の原因を見極める。すなわち、異常の原因が「不正要素の出現」であったかどうか調べ（ステップ１９０４）、そ

うでなかった場合に「不一致」との結果を返す。前記異常が「不正要素の出現」であった場合、文書構造変化規則集から適用可能な文書構造変化規則を検索する（ステップ１９０５）。ここで「適用可能」とは、現文書構造定義が文書構造変化規則の「適用先」に合致し、かつ、検査対象構造化文書に現れた不正要素が文書構造変化規則の「操作要素」に一致することを意味する。ただし、「適用先」が「＊」の場合、あらゆる文書構造定義に合致するものとする。

【００５１】

ステップ１９０５で適用可能な文書構造定義が発見（ステップ１９０６）できなかった場合、「不一致」との結果を返す。ステップ１９０５で適用可能な文書構造変化規則を発見した場合、現文書構造定義（ここでは名称）とその不一致箇所（ここでは第１の実施例の文書構造変化規則の適用場所５０２と同じ表記方法を用いて表現している）、発見した文書構造変化規則の種別の３つの情報をひとまとまりにしてスタックにＰｕｓｈし、発見した文書構造変化規則の操作要素の文書構造定義を現文書構造定義に設定して（ステップ１９０７）、検査を続行する（ステップ１９０２へ）。

ステップ１９０３で正常に終了した場合、スタックの状態を調べ（ステップ１９０８）、空の場合、検査対象構造化文書の末尾まで到達（ステップ１９１２）していれば「一致」を返して正常終了する。ステップ１９１２で検査対象構造化文書の末尾に到達していなければ「不一致」を返して終了する。ステップ１９０８でスタックが空でない場合、スタックの先頭から文書構造定義、不一致箇所、変化規則の種別をＰｏｐし、Ｐｏｐした文書構造定義を現文書構造定義に設定する（ステップ１９０９）。次に、ステップ１９０９でＰｏｐした変化規則の種別が「Ｒｅｐｌａｃｅ」であるかどうか調べ（ステップ１９１０）、「Ｒｅｐｌａｃｅ」でない場合、検査を続行する（ステップ１９０２へ）。ステップ１９０９でＰｏｐした変化規則が「Ｒｅｐｌａｃｅ」であった場合、現文書構造定義の検査位置を次の兄弟要素にする。次の兄弟要素が存在しない場合は、現要素の直後に検査位置を移動する（ステップ１９１１）。ステップ１９０９でＰｏｐした変化規則が「Ｒｅｐｌａｃｅ」でなかった場合、および、ステップ１９１１の実行後はステップ１９０２へ進み、検査を続行する。

【００５２】

図１４の構造化文書を検査対象として、図１９の手順に従って文書構造検査を行う様子を以下で示す。

【００５３】

文書構造定義集２１３には図３の「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」、図６の「ＥｎｃｒｙｐｔｅｄＤａｔａ．ｄｔｄ」、図７の「ＥｎｃｒｙｐｔｅｄＫｅｙ．ｄｔｄ」、図８の「Ｓｉｇｎａｔｕｒｅ．ｄｔｄ」、図９の「ＫｅｙＩｎｆｏ．ｄｔｄ」が、文書構造変化規則集１７１１には図１８の４つの規則が記憶されているものとする。

【００５４】

はじめに、ステップ１９０１で検査対象構造化文書（図１４の１４０１）の２行目から文書構造定義を「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」に特定する。次にステップ１９０２で、エラー発生もしくは検査終了まで通常通りの検査を行う。この例では３行目から５行目までは正常に検査が行われるが、６行目の要素「ＥｎｃｒｙｐｔｅｄＤａｔａ」まで来たところで、文書構造定義（図３）の２行目に記載されている要素「ＣｒｅｄｉｔＣａｒｄ」と一致しないためエラーが発生する。このエラーは不正要素「ＥｎｃｒｙｐｔｅｄＤａｔａ」の出現であるためステップ１９０５へ進み、文書構造変化規則集から適用先が現文書構造定義「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」と合致し、かつ、操作要素が「ＥｎｃｒｙｐｔｅｄＤａｔａ」と一致する文書構造変化規則を検索する。検索の結果、図１８に示す1番目の文書構造変化規則１８１１が見つかる。1番目の文書構造変化規則１８１１の適用先は「＊」であるが、これはどの文書構造変化規則とも合致するので「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」にも合致する。

【００５５】

次にステップ１９０７で現文書構造定義「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」、不一致箇所「／ＰｕｒｃｈａｓｅＯｒｄｅｒ／ＣｒｅｄｉｔＣａｒｄ」、発見した文書構造変化規則１の種別「Ｒｅｐｌａｃｅ」をスタックにＰｕｓｈする。その様子を図２０に示す。スタックの先頭（最上部）に「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ｜／ＰｕｒｃｈａｓｅＯｒｄｅｒ／ＣｒｅｄｉｔＣａｒｄ｜

Ｒｅｐｌａｃｅ」（２００１）が記憶される。次に、ステップ１９０５で発見した文書構造変化規則１の操作要素「ＥｎｃｒｙｐｔｅｄＤａｔａ．ｄｔｄ：／ＥｎｃｒｙｐｔｅｄＤａｔａ」を現文書構造定義に設定し、さらに検査を続ける（ステップ１９０２へ）。

【００５６】

次に、ステップ１９０２で図６の文書構造定義「ＥｎｃｒｙｐｔｅｄＤａｔａ．ｄｔｄ」６０１の「ＥｎｃｒｙｐｔｅｄＤａｔａ」要素（行番号０２）と、図１４の検査対象構造化文書１４０１の行番号０６以降の検査が行われる。本検査は図１４の検査対象構造化文書１４０１の行番号１５の終わりまで進むと、文書構造定義６０１の末尾まで到達するのでステップ１９０３で「はい」の方に進み、スタックが検査される。スタックには先ほどステップ１９０７でＰｕｓｈした情報２００１が記憶されており空ではないので、ステップ１９０９へ進み、スタックの先頭の情報２００１をＰｏｐする。この処理によりスタックは空になる。さらに、現文書構造定義をさきほどＰｏｐした文書構造定義「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」に、検査位置を「／ＰｕｒｃｈａｓｅＯｒｄｅｒ／ＣｒｅｄｉｔＣａｒｄ」に設定する。次に、ステップ１９０９でＰｏｐした文書構造変化規則の種別が「Ｒｅｐｌａｃｅ」のため、ステップ１９１１へ進む。ステップ１９１１ではステップ１９０９で「／ＰｕｒｃｈａｓｅＯｒｄｅｒ／ＣｒｅｄｉｔＣａｒｄ」に設定した現文書構造定義の検査位置を次の兄弟要素に設定する。ここではＣｒｅｄｉｔＣａｒｄ要素に次の兄弟要素が存在しないので、「ＣｒｅｄｉｔＣａｒｄ要素の直後（／ＰｕｒｃｈａｓｅＯｒｄｅｒ／ＣｒｅｄｉｔＣａｒｄ／ａｆｔｅｒ（））」へ検査位置を設定する。その後、ステップ１９０２へ進み、さらに検査を続ける。

【００５７】

ここまでで現文書構造定義「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」の「／ＰｕｒｃｈａｓｅＯｒｄｅｒ／ＣｒｅｄｉｔＣａｒｄ／ａｆｔｅｒ（）」に検査位置が進み、検査対象構造化文書（図１４の１４０１）の検査位置は行番号１５の終わりまで進んでいる。

【００５８】

ステップ１９０２で検査を続行すると、検査対象構造化文書（図１４の１４０１）の行番号１６に不正要素「ＥｎｃｒｙｐｔｅｄＫｅｙ」が出現することがわかるので、ステップ１９０５へ進み、適用先と操作要素一致する文書構造変化規則を検索する。今回の検索では、図１８に示した2番目の文書構造変化規則１８１２が見つかるのでスタックに「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ｜／ＰｕｒｃｈａｓｅＯｒｄｅｒ／ＣｒｅｄｉｔＣａｒｄ／ａｆｔｅｒ（）｜Ａｄｄ」をＰｕｓｈする。　図２１にこの時のスタックの状態を示す。

さらに、現文書構造定義を「ＥｎｃｒｙｐｔｅｄＫｅｙ．ｄｔｄ」、検査位置を「ＥｎｃｒｙｐｔｅｄＫｅｙ要素」に設定して、ステップ１９０２へ進み、検査を続ける。

今回のステップ１９０２では図１４の検査対象構造化文書１４０１の行番号２７の終わりまで検査が進んだところで、現文書構造定義「ＥｎｃｒｙｐｔｅｄＫｅｙ．ｄｔｄ」の末尾に到達するため、処理がステップ１９０８へ進む。次に、スタックの先頭の情報「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ｜2行目ＣｒｅｄｉｔＣａｒｄ要素の直後｜Ａｄｄ」をＰｏｐし、現文書構造定義および検査位置をそれぞれ「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」「／ＰｕｒｃｈａｓｅＯｒｄｅｒ／ＣｒｅｄｉｔＣａｒｄ／ａｆｔｅｒ（）」に設定する。今回Ｐｏｐした種別は「Ａｄｄ」なので、ステップ１９０２へ進み、検査を続ける。

〔００５９〕

ステップ１９０２で検査を続行すると、検査対象構造化文書の検査位置は変化しないまま、不正要素「Ｓｉｇｎａｔｕｒｅ」が出現するため、ステップ１９０５へ進み、適用先と操作要素一致する文書構造変化規則を検索する。今回の検索では、図１８に示した３番目の文書構造変化規則１８１３が見つかるのでスタックに「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ｜／ＰｕｒｃｈａｓｅＯｒｄｅｒ／ＣｒｅｄｉｔＣａｒｄ／ａｆｔｅｒ（）｜Ａｄｄ」をＰｕｓｈする。さらに、現文書構造定義を「Ｓｉｇｎａｔｕｒｅ．ｄｔｄ」、検査位置を「Ｓｉｇｎａｔｕｒｅ要素」に設定して、ステップ１９０２へ進み、検査を続ける。

〔００６０〕

今回のステップ１９０２では検査対象構造化文書１４０１の行番号４１の終わ

りまで検査が進んだところで、現文書構造定義「Ｓｉｇｎａｔｕｒｅ．ｄｔｄ」の末尾に到達するため、処理がステップ１９０８へ進む。次に、スタックの先頭の情報「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ｜／ＰｕｒｃｈａｓｅＯｒｄｅｒ／ＣｒｅｄｉｔＣａｒｄ／ａｆｔｅｒ（）｜Ａｄｄ」をＰｏｐし、現文書構造定義および検査位置をそれぞれ「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」「／ＰｕｒｃｈａｓｅＯｒｄｅｒ／ＣｒｅｄｉｔＣａｒｄ／ａｆｔｅｒ（）」に設定する。この処理によりスタックは空になる。今回Ｐｏｐした種別は「Ａｄｄ」なので、ステップ１９０２へ進み、検査を続ける。

【００６１】

ステップ１９０２で検査が、検査対象構造化文書１４０１の行番号４２の終わりまで到達すると、現文書構造定義「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」の末尾に到達するのでステップ１９０８へ進む。今回はスタックは空で、かつ、検査対象構造化文書の末尾（行番号４２の終わり）まで到達しているので、検査を終了して「一致」を返す。

【００６２】

以上で検査対象構造化文書１４０１の検査が終了する。

検査対象構造化文書１４０１の検査では、４番目の文書構造変化規則１８１４を適用しなかった。４番目の文書構造変化規則は署名対象文書が署名自体に内包される構造化文書（図１６の文書構造定義に合致する）の検査の際に利用するものである。

上記のように、本発明の第２の実施例によれば、構造化文書の全部または一部に署名が付与される、または、構造化文書の全部または一部が暗号化されるなどの理由により文書構造が変化する場合でも、また、暗号化部分を検査者が復号化できない場合でも、文書全体の構造検査を実施することが可能となる。

【００６３】

第１および第２の実施例では全ての文書構造定義を１種類の文書構造定義言語ＤＴＤで記述した場合を想定していたが、第１の実施例では「文書構造定義への参照（２２１１、２２１２）を利用してその実体を取得する時」に、また、第２の実施例では「ステップ１９０２を実行する際」に、文書構造定義の記述に利用

されている文書構造定義言語の種別を判断し、対応する文書構造検査部を選択するようにすることで、複数の文書構造定義言語を利用して文書構造定義を記述した場合でも検査対象構造化文書の構造検査を実施することができるようになる。前記「文書構造定義言語の種別の判断」は文書構造定義名で行うことができる。

【００６４】

例えば、文書構造定義名「ＰｕｒｃｈａｓｅＯｒｄｅｒ．ｄｔｄ」はその拡張子「ｄｔｄ」で、該文書構造定義がＤＴＤにより記述されていることを、また、文書構造定義名「Ｓｉｇｎａｔｕｒｅ．ｘｓｄ」はその拡張子「ｘｓｄ」で文書構造定義が「Ｗ３Ｃ　ＸＭＬ　Ｓｃｈｅｍａ」により記述されていることを意味することができる。

【００６５】

さらに、複数の文書構造定義言語を利用して文書構造定義を記述した場合でも検査対象構造化文書の構造検査を実施可能とする別の方法として、文書構造定義集２１３に記憶された文書構造定義を１種類の文書構造定義言語を用いた表現に変換する方法も適用可能である。

【００６６】

【発明の効果】

本発明により、構造化文書の全部または一部に署名が付与される、または、構造化文書の全部または一部が暗号化されるなどの理由により文書構造が変化する場合でも、暗号化部分の検査者の復号化可否に関わらず、文書全体の構造検査を実施することが可能となる。

【図面の簡単な説明】

【図１】

計算機環境を表す図である。

【図２】

実施例１の計算機内部のハードウェア構成を表す図である。

【図３】

文書構造定義の例である。

【図４】

構造化文書の例である。

【図５】

実施例１の文書構造変化規則集の例である。

【図６】

文書構造定義の例である。

【図７】

文書構造定義の例である。

【図８】

文書構造定義の例である。

【図９】

文書構造定義の例である。

【図１０】

実施例１の処理手順を表す図である。

【図１１】

文書構造変化規則１を適用した後の文書構造定義である。

【図１２】

文書構造変化規則１と２を適用した後の文書構造定義である。

【図１３】

文書構造変化規則１と２と３を適用した後の文書構造定義である。

【図１４】

構造変化後の構造化文書の例である。

【図１５】

文書構造変化規則の例である。

【図１６】

文書構造変化規則４を適用した後の文書構造定義である。

【図１７】

実施例２の計算機内部のハードウェア構成を表す図である。

【図１８】

実施例２の文書構造変化規則集の例である。

【図１９】

文書構造検査の詳細な処理手順を表す図である。

【図２０】

スタックの状態を表す図である。

【図２１】

スタックの状態を表す図である。

【図２２】

参照を利用して記述した文書構造定義の例である。

【符号の説明】

１０１・・・・・計算機、

１０２・・・・・計算機、

１０３・・・・・ネットワーク、

１０４・・・・・受信者側の計算機、

２０１・・・・・ＣＰＵ、

２０２・・・・・メモリ、

２０３・・・・・表示装置、

２０４・・・・・入力装置、

２０５・・・・・通信装置、

２０６・・・・・記憶装置、

２１１・・・・・文書構造検査部、

２１２・・・・・業務処理部、

２１３・・・・・文書構造定義集、

２１４・・・・・文書構造定義変換部、

２１５・・・・・文書構造変化規則集、

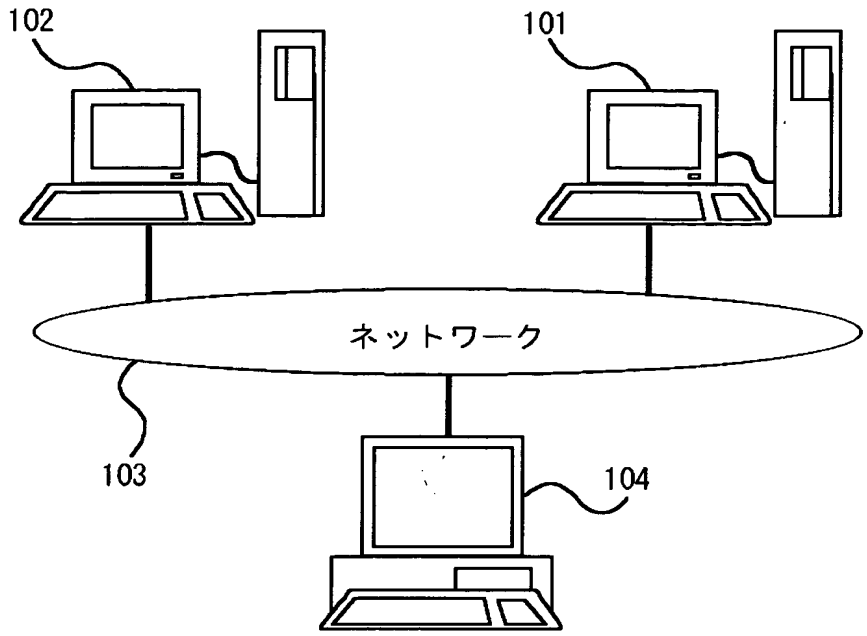
１７０１・・・・文書構造検査部、

１７１１・・・・文書構造変化規則集。

【書類名】　図面

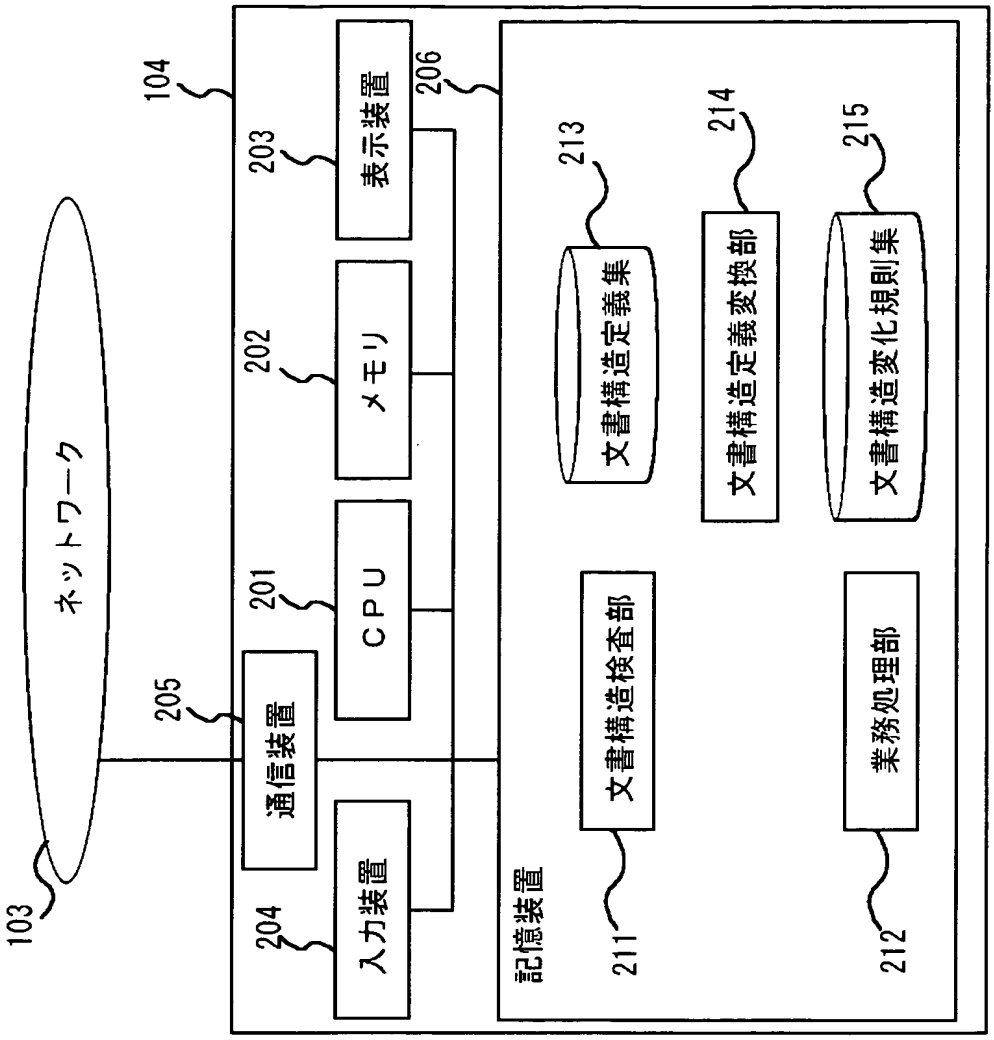
【図１】

図　1



【図２】

図　2


ネットワーク
103
104
通信装置
205
CPU
201
メモリ
202
表示装置
203
入力装置
204
206
記憶装置
文書構造検査部
211
文書構造定義集
213
文書構造定義変換部
214
業務処理部
212
文書構造変化規則集
215

【図３】

図　３

```
01  <!DOCTYPE PurchaseOrder[
02    <!ELEMENT PurchaseOrder (UserID, Price, CreditCard )>
03      <!ATTLIST PurchaseOrder Id ID #IMPLIED>
04    <!ELEMENT UserID (#PCDATA)>
05    <!ELEMENT Price (#PCDATA)>
06    <!ELEMENT CreditCard (Issure, Number, Expire, Owner)>
07    <!ELEMENT Issure (#PCDATA)>
08    <!ELEMENT Number (#PCDATA)>
09    <!ELEMENT Expire (#PCDATA)>
10    <!ELEMENT Owner (#PCDATA)>
11]>
```

320

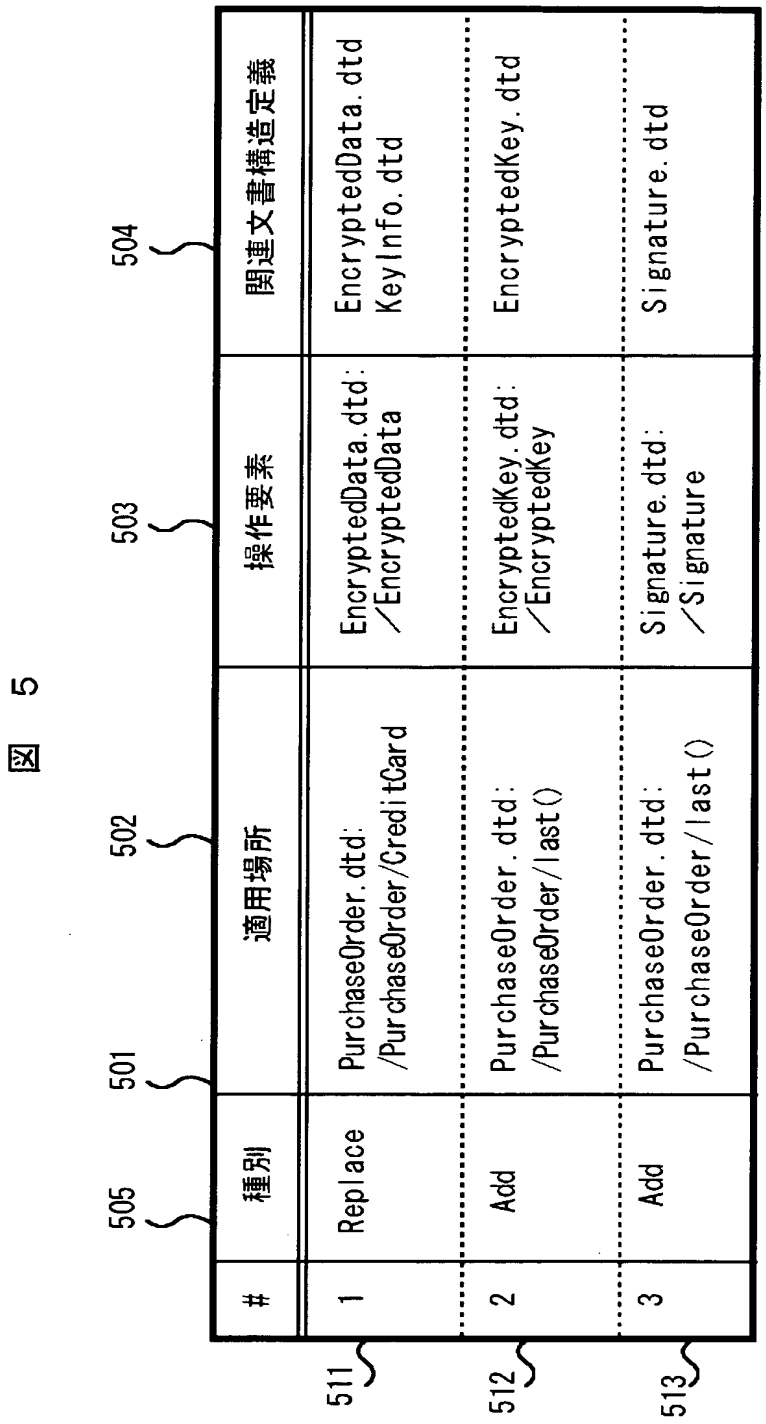
【図４】

図　４

```
01  <?Xml version="1.0"?>
02  <!DOCTYPE PurchaseOrder SYSTEM "PurchaseOrder.dtd">
03  <PurchaseOrder>
04    <UserID>10194970</UserID>
05    <Price>100000</Price>
06    <CreditCard>
07    <Issur>SDL</Issuer>
08    <Number>1234-5678-9012-3456</Number>
09    <Expire>12/05</Expire>
10    <Owner>Larry Gates</Owner>
11   </CreditCard>
12  </PurchaseOrder>
```

【図５】

図 5


505
501
502
503
504
#
種別
適用場所
操作要素
関連文書構造定義
1
Replace
PurchaseOrder.dtd:
/PurchaseOrder/CreditCard
EncryptedData.dtd:
／EncryptedData
EncryptedData.dtd
KeyInfo.dtd
2
Add
PurchaseOrder.dtd:
/PurchaseOrder/last()
EncryptedKey.dtd:
／EncryptedKey
EncryptedKey.dtd
3
Add
PurchaseOrder.dtd:
/PurchaseOrder/last()
Signature.dtd:
／Signature
Signature.dtd
511
512
513

【図６】

図　６

601

```
01 <!DOCTYPE EncryptedData[
02 <!ELEMENT Encrypted Data(EncryptionMethod,KeyInfo,CipherData)>
03    <!ATTLIST EncryptedData Id ID #REQUIRED>
04 <!ELEMENT EncryptionMethod(#PCDATA)>
05    <!ATTLIST EncryptionMethod Algorithm CDATA #REQUIRED>
06 <!ELEMENT CipherData(CipherValue)>
07 <!ELEMENT CipherValue(#PCDATA)>
08 ]>
```

【図７】

図　7

```
01<!DOCTYPE EncryptedKey [
02  <!ELEMENT EncryptedKey(EncryptionMethod,KeyInfo,CipherData,ReferenceList>
03    <!ATTLIST EncryptedKey Id ID #REQUIRED>
04  <!ELEMENT ReferenceList(DataReference | Key Reference)+>
05  <!ELEMENT DataReference(#PCDATA)>
06    <!ATTLIST DataReference URI CDATA #REQUIRED>
07  <!ELEMENT KeyReference(#PCDATA)>
08    <!ATTLIST KeyReference URI CDATA #REQUIRED>
09 ]>
```

【図８】

図　8

```
01<!DOCTYPE Signature [
02  <!ELEMENT  Signature(SignedInfo,SignatureValue,KeyInfo?)>
03  <!ELEMENT  SignedInfo(CanonicalizationMethod,SignatureMetod,Reference+)>
04  <!ELEMENT  CanonicalizationMethod(#PCDATA)>
05    <!ATTLIST CanonicalizationMetod Algorithm CDATA #REQUIRED>
06  <!ELEMENT  SignatureMethod(#PCDATA)>
07    <!ATTLIST SignatureMethod Algorithm CDATA #REQUIRED>
08  <!ELEMENT  Reference(DigestMethod,DigestValue)>
09    <!ATTLIST Reference URI CDATA #REQUIRED>
10  <!ELEMENT  DigestMethod(#PCDATA)>
11    <!ATTLIST DigestMethod Algorithm CDATA #REQUIRED>
12  <!ELEMENT  DigestValue(#PCDATA)>
13  <!ELEMENT  SignatureValue(#PCDATA)>
14 ]>
```

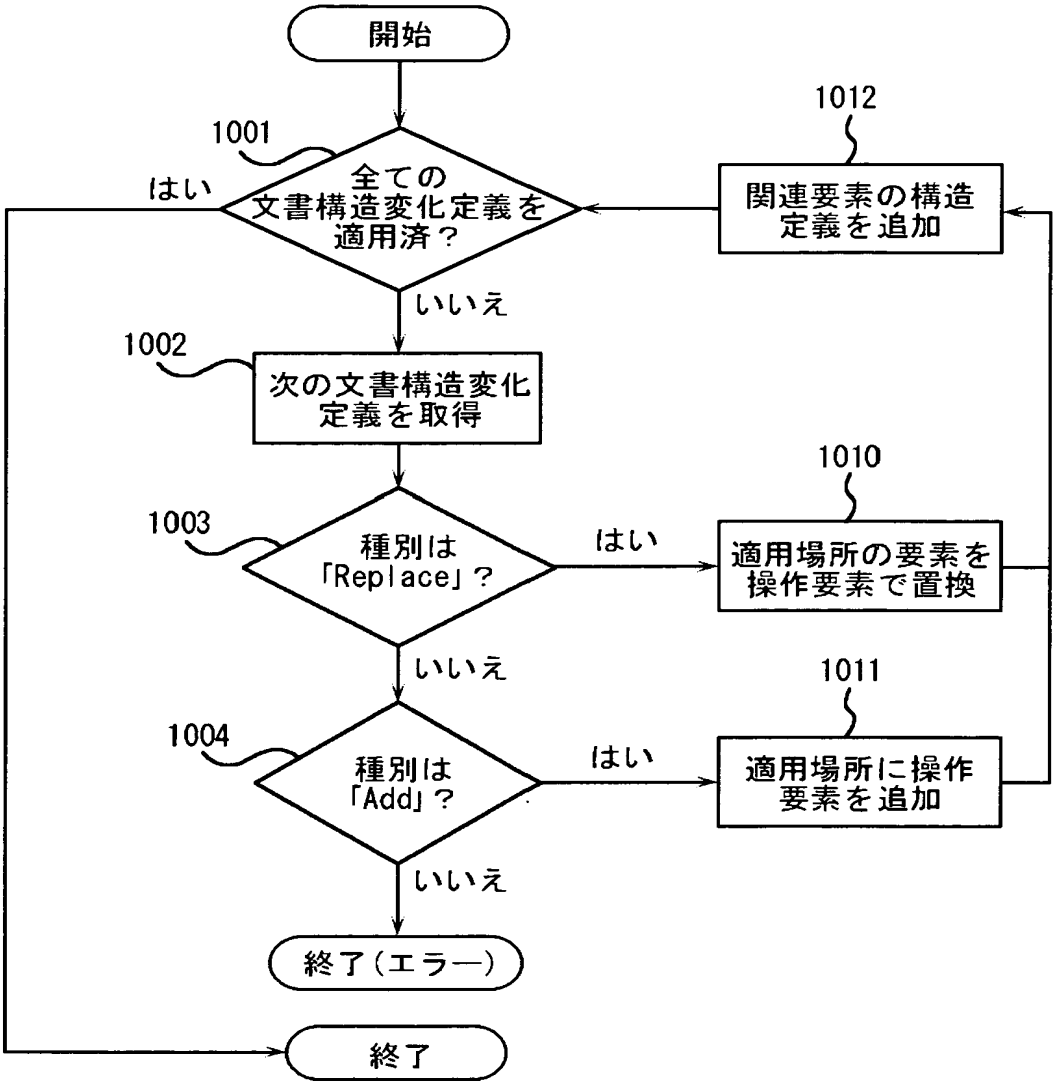
【図９】

図　９

```
01<!DOCTYPE KeyInfo[
02  <!ELEMENT KeyInfo(RetrievalMethod | KeyName)>
03  <!ELEMENT RetrievalMethod(#PCDATA)>
04    <!ATTLIST RetrievalMethod
05          Type        CDATA       #REQUIRED
06          URI         CDATA       #REQUIRED>
07  <!ELEMENT KeyName(#PCDATA)>
08 ]>
```

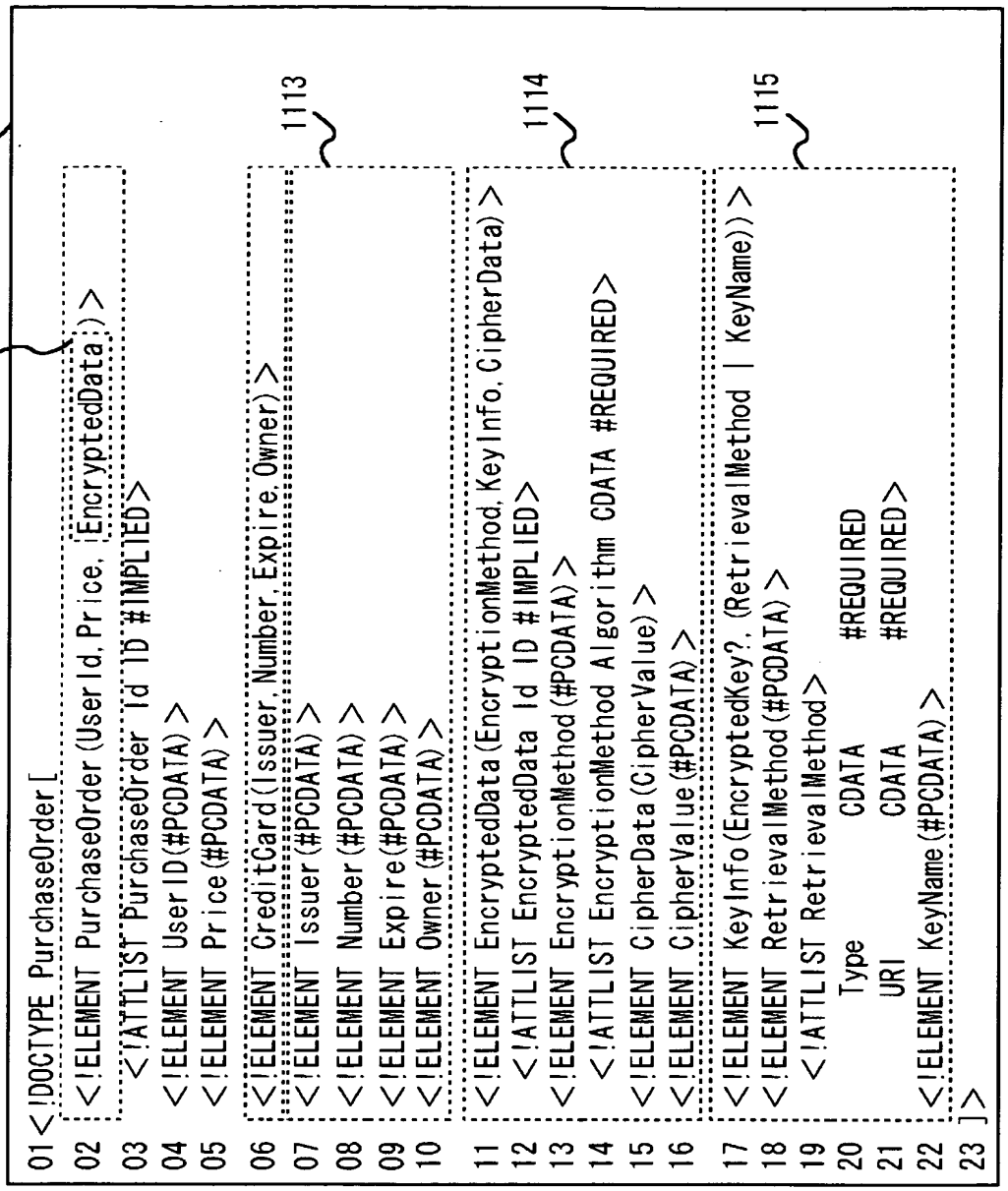
【図１０】

図　１０


開始
1001
全ての
文書構造変化定義を
適用済？
はい
いいえ
1012
関連要素の構造
定義を追加
1002
次の文書構造変化
定義を取得
1003
種別は
「Replace」？
はい
1010
適用場所の要素を
操作要素で置換
いいえ
1004
種別は
「Add」？
はい
1011
適用場所に操作
要素を追加
いいえ
終了（エラー）
終了

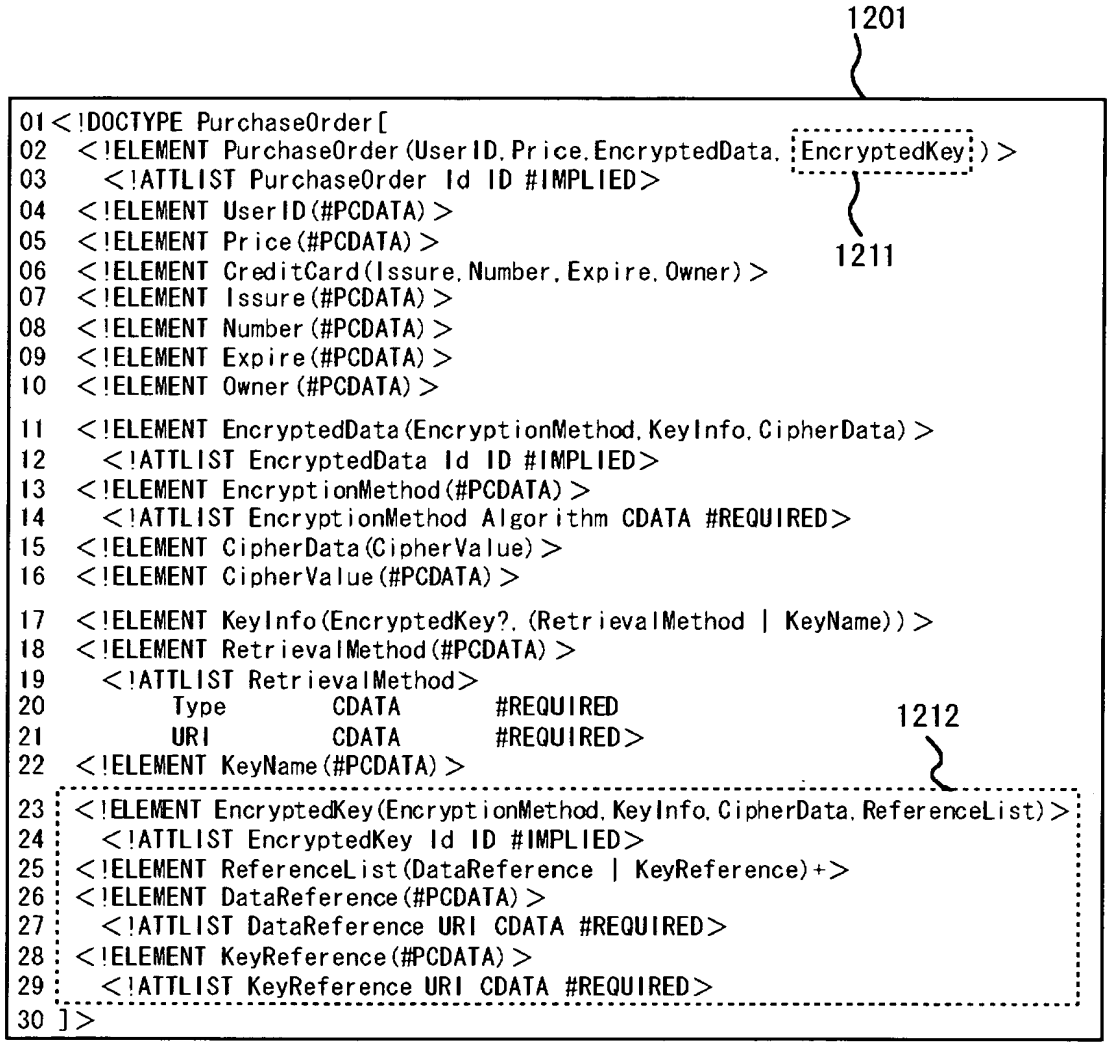
【図１１】

図　１１


1101
1121
1113
1114
1115
01 <!DOCTYPE PurchaseOrder[
02 <!ELEMENT PurchaseOrder (UserId, Price, EncryptedData)>
03 <!ATTLIST PurchaseOrder Id ID #IMPLIED>
04 <!ELEMENT UserID (#PCDATA)>
05 <!ELEMENT Price (#PCDATA)>
06 <!ELEMENT CreditCard (Issuer, Number, Expire, Owner)>
07 <!ELEMENT Issuer (#PCDATA)>
08 <!ELEMENT Number (#PCDATA)>
09 <!ELEMENT Expire (#PCDATA)>
10 <!ELEMENT Owner (#PCDATA)>
11 <!ELEMENT EncryptedData (EncryptionMethod, KeyInfo, CipherData)>
12 <!ATTLIST EncryptedData Id ID #IMPLIED>
13 <!ELEMENT EncryptionMethod (#PCDATA)>
14 <!ATTLIST EncryptionMethod Algorithm CDATA #REQUIRED>
15 <!ELEMENT CipherData (CipherValue)>
16 <!ELEMENT CipherValue (#PCDATA)>
17 <!ELEMENT KeyInfo (EncryptedKey?, (RetrievalMethod | KeyName))>
18 <!ELEMENT RetrievalMethod (#PCDATA)>
19 <!ATTLIST RetrievalMethod>
20 Type CDATA #REQUIRED
21 URI CDATA #REQUIRED>
22 <!ELEMENT KeyName (#PCDATA)>
23]>

【図１２】

図　１２


1201
1211
1212
01 <!DOCTYPE PurchaseOrder[
02 <!ELEMENT PurchaseOrder(UserID, Price, EncryptedData, EncryptedKey)>
03 <!ATTLIST PurchaseOrder Id ID #IMPLIED>
04 <!ELEMENT UserID(#PCDATA)>
05 <!ELEMENT Price(#PCDATA)>
06 <!ELEMENT CreditCard(Issure, Number, Expire, Owner)>
07 <!ELEMENT Issure(#PCDATA)>
08 <!ELEMENT Number(#PCDATA)>
09 <!ELEMENT Expire(#PCDATA)>
10 <!ELEMENT Owner(#PCDATA)>
11 <!ELEMENT EncryptedData(EncryptionMethod, KeyInfo, CipherData)>
12 <!ATTLIST EncryptedData Id ID #IMPLIED>
13 <!ELEMENT EncryptionMethod(#PCDATA)>
14 <!ATTLIST EncryptionMethod Algorithm CDATA #REQUIRED>
15 <!ELEMENT CipherData(CipherValue)>
16 <!ELEMENT CipherValue(#PCDATA)>
17 <!ELEMENT KeyInfo(EncryptedKey?, (RetrievalMethod | KeyName))>
18 <!ELEMENT RetrievalMethod(#PCDATA)>
19 <!ATTLIST RetrievalMethod>
20 Type CDATA #REQUIRED
21 URI CDATA #REQUIRED>
22 <!ELEMENT KeyName(#PCDATA)>
23 <!ELEMENT EncryptedKey(EncryptionMethod, KeyInfo, CipherData, ReferenceList)>
24 <!ATTLIST EncryptedKey Id ID #IMPLIED>
25 <!ELEMENT ReferenceList(DataReference | KeyReference)+>
26 <!ELEMENT DataReference(#PCDATA)>
27 <!ATTLIST DataReference URI CDATA #REQUIRED>
28 <!ELEMENT KeyReference(#PCDATA)>
29 <!ATTLIST KeyReference URI CDATA #REQUIRED>
30]>

【図１３】

図　13

```
01 <!DOCTYPE PurchaseOrder[
02   <!ELEMENT PurchaseOrder (UserID,Price,EncryptedData,EncryptedKey, Signature?)>
03     <!ATTLIST PurchaseOrder Id ID #IMPLIED>
04   <!ELEMENT UserId(#PCDATA)>
05   <!ELEMENT Price(#PCDATA)>
06   <!ELEMENT CreditCard(Issure,Number,Expire,Owner)>
07   <!ELEMENT Issure(#PCDATA)>
08   <!ELEMENT Number(#PCDATA)>
09   <!ELEMENT Expire(#PCDATA)>
10   <!ELEMENT Owner(#PCDATA)>
11   <!ELEMENT EncryptedData(EncryptionMethod,KeyInfo,CipherData)>
12     <!ATTLIST EncryptdData Id ID #IMPLIED>
13   <!ELEMENT EncryptionMethod(#PCDATA)>
14     <!ATTLIST EncryptionMethod Algorithm CDATA #REQUIRED>
15   <!ELEMENT CipherData(CipherValue)>
16   <!ELEMENT CipherValue(#PCDATA)>
17   <!ELEMENT KeyInfo(EncryptedKey?, (RetrievalMethod | KeyName))>
18   <!ELEMENT RetrievalMethod(#PCDATA)>
19     <!ATTLIST RetrievalMethod
20           Type        CDATA      #REQUIRED
21           URI         CDATA      #REQUIRED>
22   <!ELEMENT KeyName(#PCDATA)>
23   <!ELEMENT EncryptedKey(EncryptionMethod,KeyInfo,CipherData,ReferenceList)>
24     <!ATTLIST EncryptedKey Id ID #MPLIED>
25   <!ELEMENT ReferenceList(DataReference | KeyReference)+>
26   <!ELEMENT DataReference(#PCDATA)>
27     <!ATTLIST DataReference URI CDATA #REQUIRED>
28   <!ELEMENT KeyReference(#PCDATA)>
29     <!ATTLIST KeyReference URI CDATA #REQUIRED>
30   <!ELEMENT Signature(SignedInfo,SignatureValue,KeyInfo?)>
31   <!ELEMENT SignedInfo(CanonicalizationMethod,SignatureMethod,Reference+)>
32   <!ELEMENT CanonicalizationMethod(#PCDATA)>
33     <!ATTLIST CanonicalizationMethod Algorithm CDATA #REQUIRED>
34   <!ELEMENT SignatureMethod(#PCDATA)>
35     <!ATTLIST SignatureMethod Algorithm CDATA #REQUIRED>
36   <!ELEMENT Reference(DigesMethod,DigestValue)>
37     <!ATTLIST Reference URI CDATA #REQUIRED>
38   <!ELEMENT DigestMethod(#PCDATA)>
39     <!ATTLIST DigestMethod Algorithm CDATA #REQUIRED>
40   <!ELEMENT DigestValue(#PCDATA)>
41   <!ELEMENT SignatureValue(#PCDATA)>
42 ]>
```

1311

1312

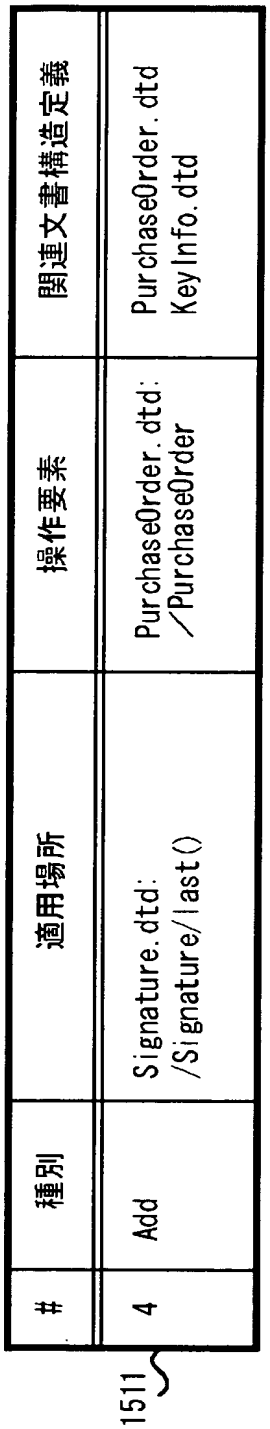
【図１４】

図　１４

```
01<?Xml version="1.0"?>
02<!DOCTYPE PurchaseOrder SYSTEM "PurchaseOrder.dtd">
03<PurchaseOrder Id="po">                                    1401
04    <UserID>10194970</UserID>
05    <Price>100000</Price>
06    <EncryptedData Id="poED">                              1411
07      <EncryptionMethod Algorithm="http://www.w3.org/xmlenc#aes128"/>
08      <KeyInfo>
09        <RetrievalMethod Type="http://www.w3.org/xmlence#EncryptedKey"
10          URI="#opEK"/>
11      </KeyInfo>
12      <CipherData>
13        <CipherValue>SrYKz0a6iu/gi.....y5UZhTTaY9</CipherValue>
14      </CipherData>
15    </EncryptedData>
16    <EncryptedKey Id="poEK">                               1412
17      <EncrypitionMethod Algorithm="http://www.w3.org/xmlenc#rsa"/>
18      <KeyInfo>
19        <KeyName>poWrapKey</KeyName>
20      </KeyInfo>
21      <CipherData>
22        <CipherValue>kjZVmUbShov4v.....wqbYwQri7QH</CipherValue>
23      </CipherData>
24      <ReferenceList>
25        <DataReference URi="#poED"/>
26      </ReferenceList>
27    </EncryptedKey>
28    <Signature>                                            1413
29      <SignedInfo>
30        <CanonicalizationMethod Algorithm ="http:/www.w3.org/xml#c14n"/>
31        <SignatureMethod Algorithm ="http://www.w3.org/xmldsg#rsa-sha1"/>
32        <Reference URI="#po">
33          <DigestMethod Algorithm ="http://www.w3.org/xmldsig#sha1"/>
34          <DigestValue>AZA0VqTorSSJ70BCA/tLY93rFMs=</DigestValue>
35        </Reference>
36      </SignedInfo>
37      <SignatureValue>ZknU0aJsxNR5.....InHhiG25PKg==</SignatureValue>
38      <KeyInfo>
39        <KeyName>Hitachi.SDL</KeyName>
40      </KeyInfo>
41    </Signature>
42</PurchaseOrder>
```

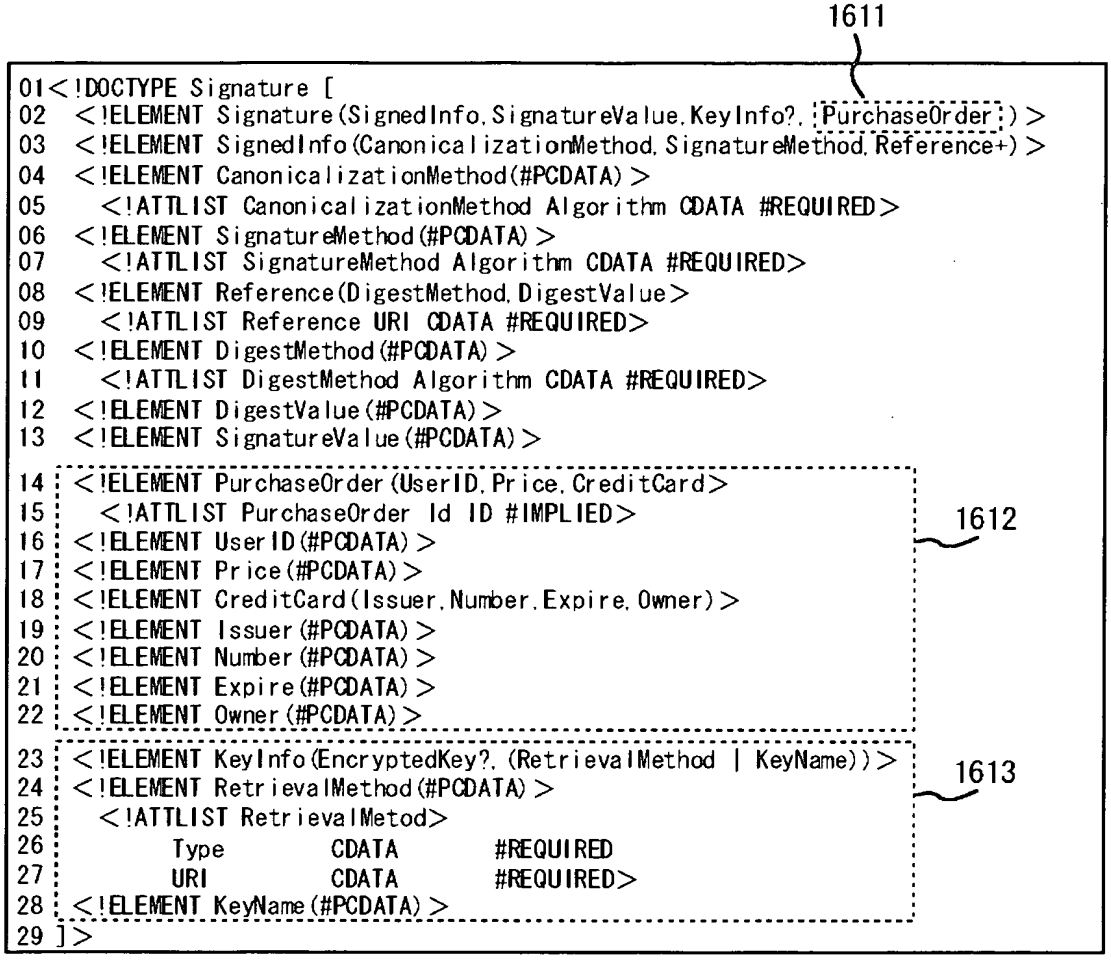
【図１５】

図　１５


#
種別
適用場所
操作要素
関連文書構造定義
4
Add
Signature.dtd:
/Signature/last()
PurchaseOrder.dtd:
／PurchaseOrder
PurchaseOrder.dtd
KeyInfo.dtd
1511

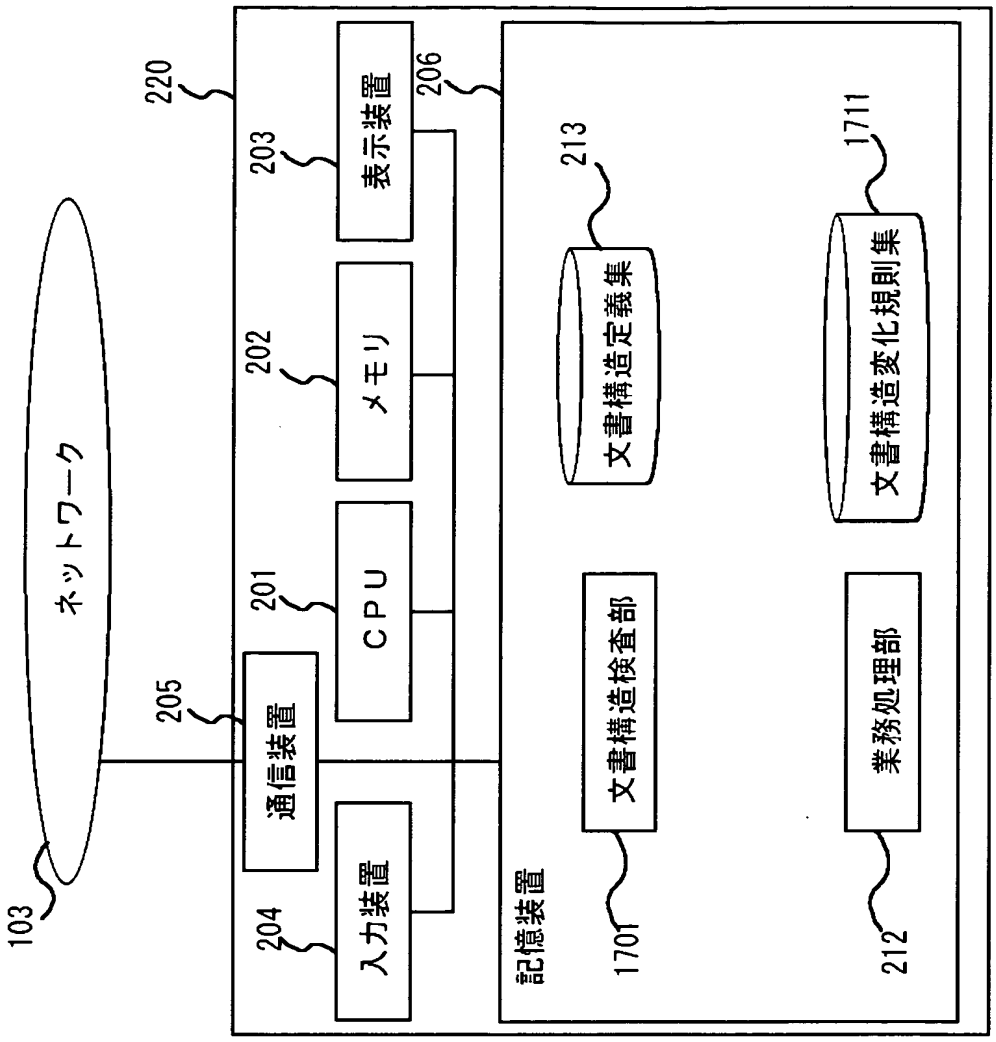
【図１６】

図　１６


1611
01<!DOCTYPE Signature [
02 <!ELEMENT Signature(SignedInfo,SignatureValue,KeyInfo?,PurchaseOrder)>
03 <!ELEMENT SignedInfo(CanonicalizationMethod,SignatureMethod,Reference+)>
04 <!ELEMENT CanonicalizationMethod(#PCDATA)>
05 <!ATTLIST CanonicalizationMethod Algorithm CDATA #REQUIRED>
06 <!ELEMENT SignatureMethod(#PCDATA)>
07 <!ATTLIST SignatureMethod Algorithm CDATA #REQUIRED>
08 <!ELEMENT Reference(DigestMethod,DigestValue>
09 <!ATTLIST Reference URI CDATA #REQUIRED>
10 <!ELEMENT DigestMethod(#PCDATA)>
11 <!ATTLIST DigestMethod Algorithm CDATA #REQUIRED>
12 <!ELEMENT DigestValue(#PCDATA)>
13 <!ELEMENT SignatureValue(#PCDATA)>
14 <!ELEMENT PurchaseOrder(UserID,Price,CreditCard>
15 <!ATTLIST PurchaseOrder Id ID #IMPLIED>
16 <!ELEMENT UserID(#PCDATA)>
17 <!ELEMENT Price(#PCDATA)>
18 <!ELEMENT CreditCard(Issuer,Number,Expire,Owner)>
19 <!ELEMENT Issuer(#PCDATA)>
20 <!ELEMENT Number(#PCDATA)>
21 <!ELEMENT Expire(#PCDATA)>
22 <!ELEMENT Owner(#PCDATA)>
1612
23 <!ELEMENT KeyInfo(EncryptedKey?,(RetrievalMethod | KeyName))>
24 <!ELEMENT RetrievalMethod(#PCDATA)>
25 <!ATTLIST RetrievalMetod>
26 Type CDATA #REQUIRED
27 URI CDATA #REQUIRED>
28 <!ELEMENT KeyName(#PCDATA)>
1613
29]>

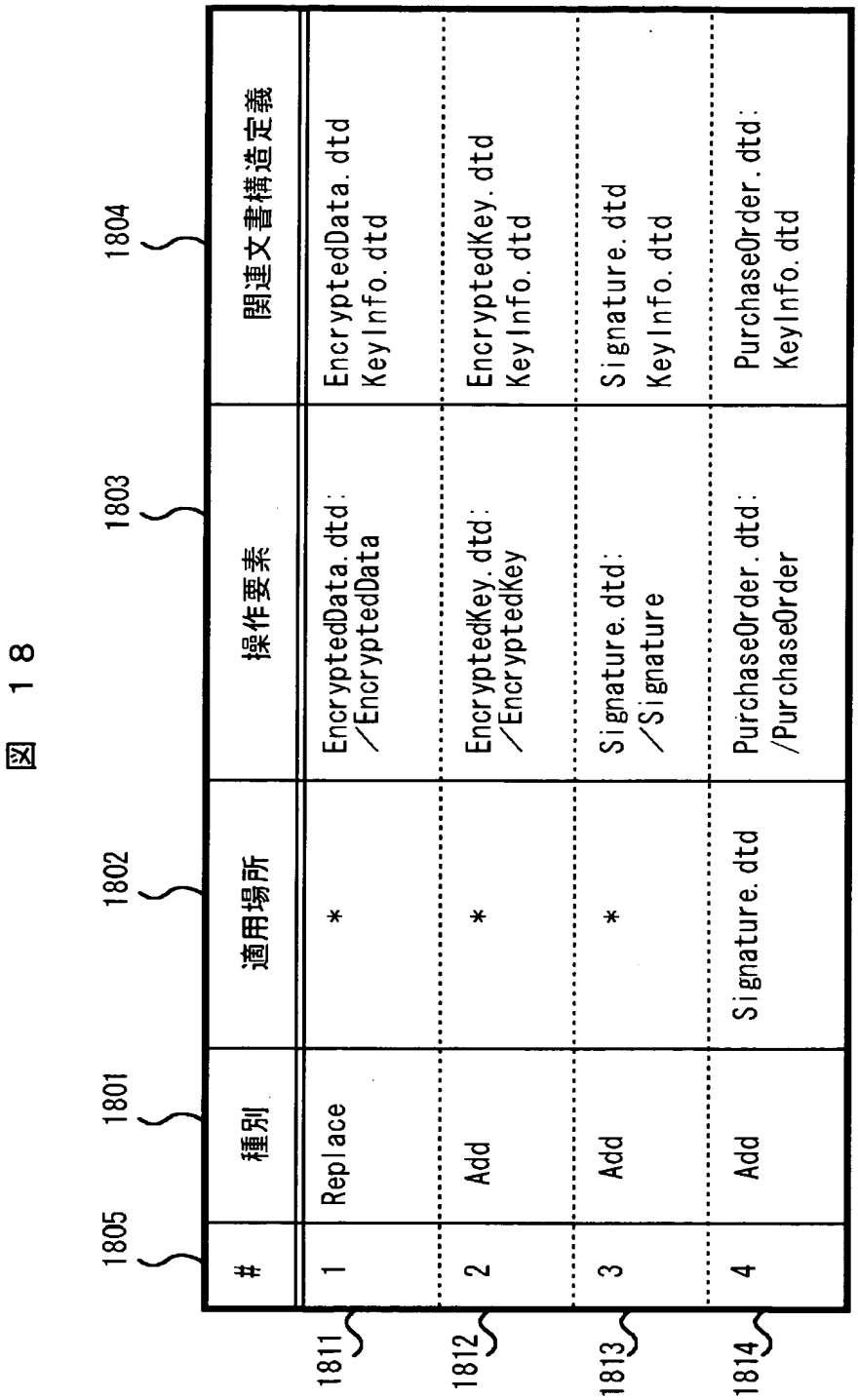
【図１７】

図　１７


ネットワーク
103
220
通信装置
205
CPU
201
メモリ
202
表示装置
203
入力装置
204
206
記憶装置
文書構造検査部
1701
業務処理部
212
文書構造定義集
213
文書構造変化規則集
1711

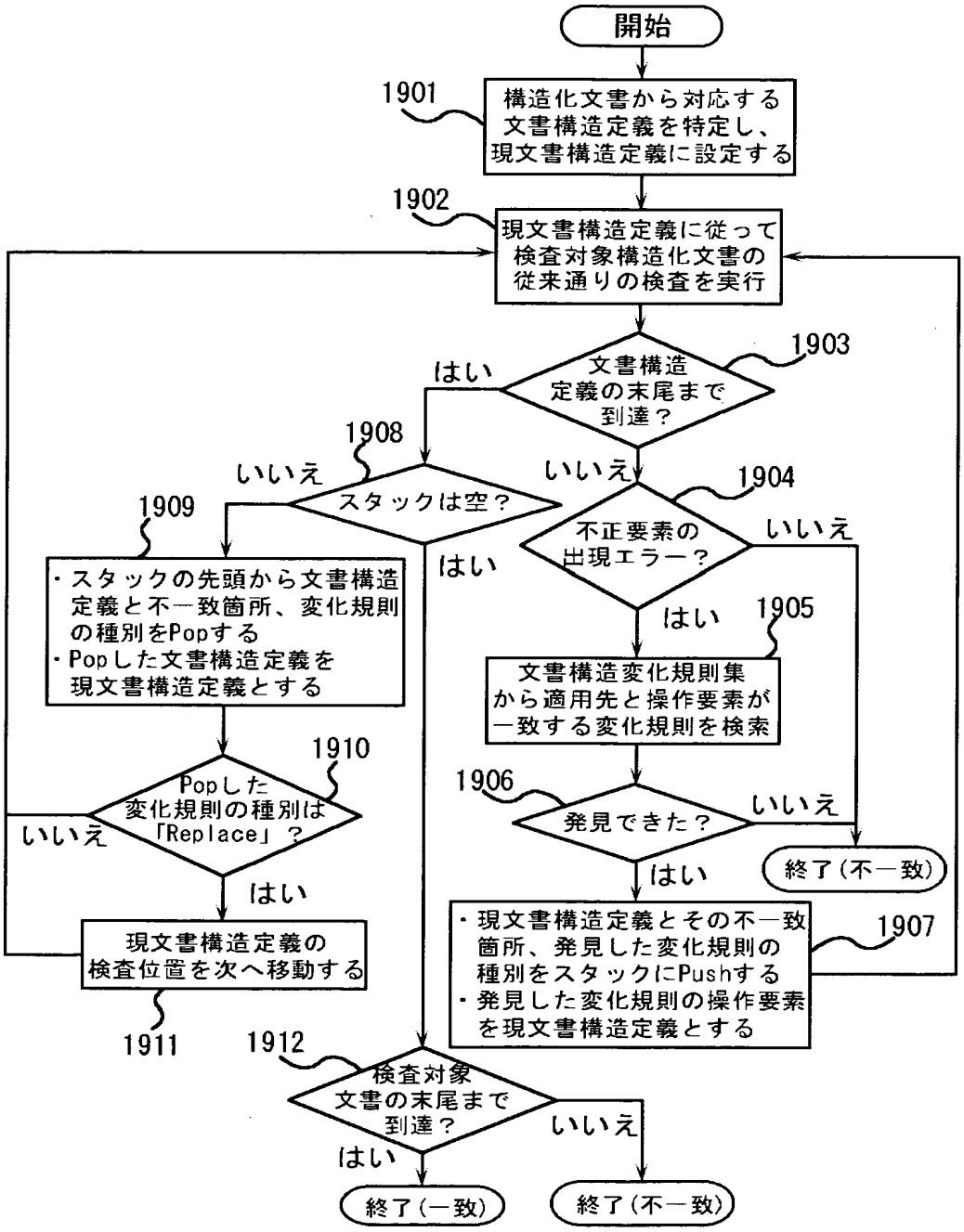
【図１８】

図　１８


1805
1801
1802
1803
1804
#
種別
適用場所
操作要素
関連文書構造定義
1
Replace
*
EncryptedData.dtd:
／EncryptedData
EncryptedData.dtd
KeyInfo.dtd
2
Add
*
EncryptedKey.dtd:
／EncryptedKey
EncryptedKey.dtd
KeyInfo.dtd
3
Add
*
Signature.dtd:
／Signature
Signature.dtd
KeyInfo.dtd
4
Add
Signature.dtd
PurchaseOrder.dtd:
/PurchaseOrder
PurchaseOrder.dtd:
KeyInfo.dtd
1811
1812
1813
1814

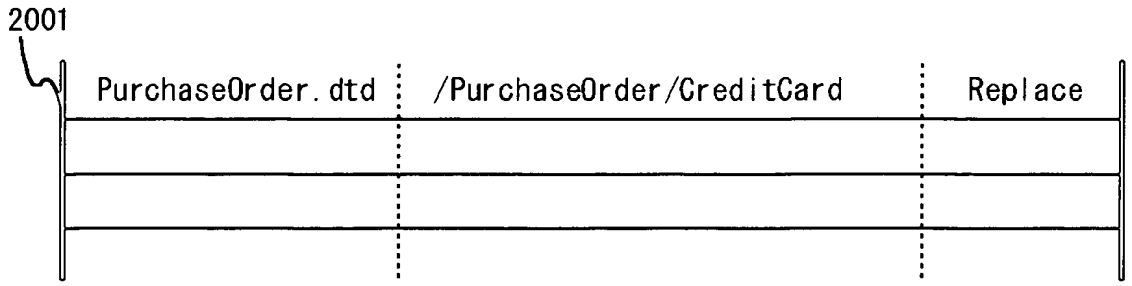
【図１９】

図　１９


開始
1901
構造化文書から対応する文書構造定義を特定し、現文書構造定義に設定する
1902
現文書構造定義に従って検査対象構造化文書の従来通りの検査を実行
1903
文書構造定義の末尾まで到達？
はい
1908
いいえ
スタックは空？
1909
・スタックの先頭から文書構造定義と不一致箇所、変化規則の種別をPopする
・Popした文書構造定義を現文書構造定義とする
1910
Popした変化規則の種別は「Replace」？
いいえ
はい
現文書構造定義の検査位置を次へ移動する
1911
いいえ
1904
不正要素の出現エラー？
いいえ
はい
1905
文書構造変化規則集から適用先と操作要素が一致する変化規則を検索
1906
発見できた？
いいえ
終了（不一致）
はい
・現文書構造定義とその不一致箇所、発見した変化規則の種別をスタックにPushする
・発見した変化規則の操作要素を現文書構造定義とする
1907
1912
検査対象文書の末尾まで到達？
はい
いいえ
終了（一致）
終了（不一致）

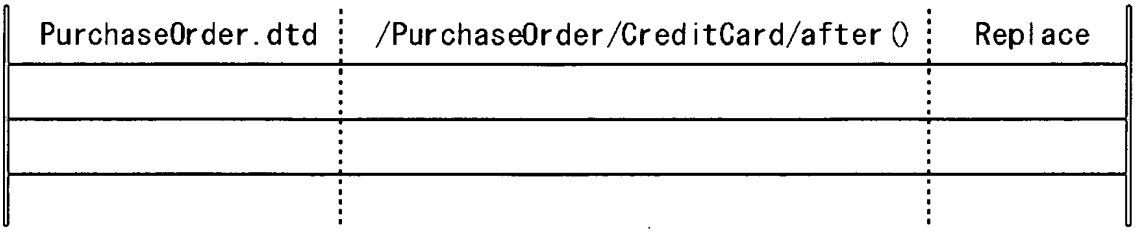
【図２０】

図　２０



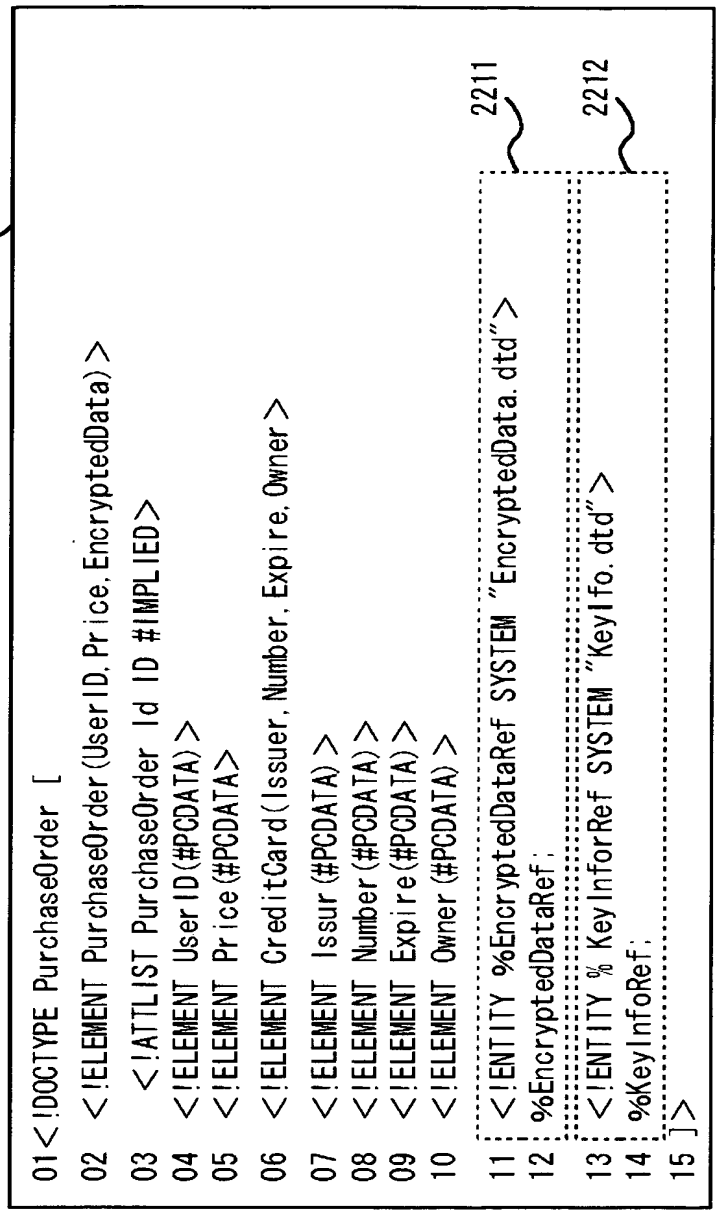
【図２１】

図　２１



【図２２】

図　２２


2201
01<!DOCTYPE PurchaseOrder [
02 <!ELEMENT PurchaseOrder(UserID,Price,EncryptedData)>
03 <!ATTLIST PurchaseOrder Id ID #IMPLIED>
04 <!ELEMENT UserID(#PCDATA)>
05 <!ELEMENT Price(#PCDATA>
06 <!ELEMENT CreditCard(Issuer,Number,Expire,Owner>
07 <!ELEMENT Issur(#PCDATA)>
08 <!ELEMENT Number(#PCDATA)>
09 <!ELEMENT Expire(#PCDATA)>
10 <!ELEMENT Owner(#PCDATA)>
11 <!ENTITY %EncryptedDataRef SYSTEM "EncryptedData.dtd">
12 %EncryptedDataRef;
13 <!ENTITY % KeyInforRef SYSTEM "KeyIfo.dtd">
14 %KeyInfoRef;
15]>
2211
2212

【書類名】　要約書

【要約】

【課題】

構造化文書に署名付与、暗号化などの処理を施すと文書の構造が変化するため、署名付与、暗号化などの処理を施さない文書と同じ方法で構造検査を実施することができない。

【解決手段】

文書に対する署名付与の場所と方式と署名データの構造、暗号化対象のデータと暗号化方式など文書構造を変化させる処理の内容をすべて事前に取り決め、文書構造検査時に利用される文書構造定義を検査前に署名付与／暗号化後の形式に変換し、該変換後の文書構造定義を利用して文書構造検査を実施する。

本発明により、構造化文書の全部または一部に署名が付与される、または、構造化文書の全部または一部が暗号化されるなどの理由により文書構造が変化する場合でも、暗号化部分の検査者の復号化可否に関わらず、文書全体の構造検査を実施することが可能となる。

【選択図】　図２

特願２００３－１４０５６８

出　願　人　履　歴　情　報

識別番号　［０００００５１０８］

１．変更年月日　１９９０年　８月３１日
［変更理由］　新規登録
住　所　東京都千代田区神田駿河台４丁目６番地
氏　名　株式会社日立製作所